VISCOUS INCOMPRESSIBLE FLOW COMPUTATIONS FOR 3-D STEADY AND UNSTEADY FLOWS

Dochan Kwak
NASA Ames Research Center
Moffett Field, CA

Lectures Presented
at
School of Mechanical and Aerospace Engineering
Seoul National University
October 8-12, 2001
Seoul, Korea

ALL the material presented in this lecture has been widely disseminated in many publications and presentations within US and abroad. Major publications/presentations are listed below:

- Methods

  Kwak, D., 'Computation of Viscous Incompressible Flows,' *von Karman Institute for Fluid Dynamics, Lecture Series* 1989-04, Mar. 6-10,1989. (Also NASA TM 101090, March 1989)

  Kwak, D., CFD Short Course - AGARD Support Project, Technical University of Lisbon, Jun. 17-19, 1996

  AIAA Paper 84-0253, AIAA 22nd Aerospace Sciences Meeting, January 9-12, 1984, Reno, Nevada (AIAA J., vol 24, No. 3, 390-396, Mar. 1986

  Rogers, S. E. and Kwak, D., 'An Upwind Differencing Scheme for the Time-Accurate Incompressible Navier-Stokes Equations,' AIAA 88-2583, AIAA 6th Applied Aerodynamics Conference, June 6-8, 1988. (AIAA J. vol. 28 No. 2, pp 253-262, February, 1990)

  Kiris, C., and Kwak, D., "Numerical Solution of Incompressible Navier-Stokes Equations Using a Fractional-Step Approach *Computers & Fluids*, in press, 2000.

- Applications

  The First MIT Conference on Computational Fluid and Solid Mechanics, Cambridge MA, June 12-14, 2001, entitled as "High-End Computing for Incompressible Flows"

  The Sixth U.S. National Congress on Computational Mechanics, Dearborn, MI August 1-4, 2001, entitled as "Computational Hemodynamics Involving Mechanical Devices"

- Discussion on general CFD challenges

  Perspectives on the Future of CFD, FLUIDS 2000 Conference, Denver, Cobrado, June 19-22, 2000

Acknowledgement

Drs. Cetin Kiris and Stuart Rogers have contributed much of the material presented here.

Numerous other researchers at NASA Ames Research Center have worked on incompressible flow methods development in the past:
Jennifer Dsacles-Mariani, Moshe Rosenfeld, Seokkwan Yoon, Jong-Youb Sa, and Marcel Vinokur

Design of Lecture

It is intended to present incompressible flow CFD from the real-world applications point of view. It will be organized in three parts:

Part 1 : Basics
Formulations, solution methods, and some historical notes

Part 2 : Applications
If the value of CFD tools is viewed from engineering practices, there are issues beyond flow solution methods. The process is discussed using following two examples:

- Turbopump
- Biofluid

Part 3 : Discussion
Discussion/seminar on "Current Topics in CFD"
Much discussion has been on-going whether CFD is a "mature technology"
A personal view on the state-of-the-art in CFD will be discussed

Outline of Presentation

- Introduction
- Review of Solution Approach
- Artificial Compressibility Method
  - Steady-state formulation
  - Time-accurate formulation
- Pressure Projection Method
- Solver/Tool development

- Problem Solving Procedures
  - Historical example
  - Some building block studies
  - Parallel Implementation
  - Turbopump flow
- Biofluid

- Discussion/seminar on "Current Topics in CFD"

Introduction

- Background
  - Flow devices tend to be compact and highly efficient
  - Experimental approach can be expensive, and model tests need to be extrapolated
  - CFD may offer an alternative to reduce cost/time for development

- Scope of Presentation
  - CFD is viewed as an engineering tool
  - Will review solution methods for incompressible N-S equations with and emphasis on 3-D applications
  - Will discuss numerical/physical characteristics of primitive variable approach
  - Solution processes are discussed using real examples

Introduction

- Role of CFD

  Past

  Functionality was the most important aspect:

  Primary concern was "performace" prediction

  Current interest

  For engineering applications:

  "Cost" and "safety" are big issues-CFD is being used to reduce design cycle time and to expand safe operating envelope

  - CFD is being used routinely in aerodynamic design of commercial aircraft

    ⇒ Return on CFD research is viewed as only incremental
  - Design of aircraft engine components is validated by CFD

    ⇒ CFD-based design of an entire engine is yet to be realized
  - Virtual flight/maneuver and systems analysis require generation of large data sets

    ⇒ Need CFD+IT : high-fidelity CFD solutions are still expensive

  For research (e.g. flow physics study or new enabling technology):

  "Idea" is the key, and can be explored via "big" problem

Viscous Incompressible Flow

- Formulation

  Can be viewed as a limiting case of compressible flow where the flow speed is insignificant compared to the speed of sound

  - Low speed aerodynamics ⇒ Preconditioned compressible N-S eq.

  Or truly incompressible

  - Hydrodynamics

- Some Examples
  - Components of liquid rocket engine
  - Hydrodynamics (Submarines, propellers, ...)
  - Ground vehicles (automobile aerodynamics, internal flows...)
  - Biofluid problems (hemodynamics, artificial heart, lung, ...)
  - Some Earth Science problems (with variable density)

Solution Methods for Incompressible N-S Equations

- Time-integration scheme:

  Primitive Variable Methods

Based on Compressible Flow Algorithm

- Artificial Compressibility Method (Chorin, 1967; Temam, 1977)

  ADI Scheme, FD (Central+diss) (Beam& Warming, 1978; Briley-McDonald, 1977)

  LU-SGS, FV (Central+diss) (Yoon...., 1987...)

  Line Relaxation, FD (Upwind) (... MacCormack, 1985)

  GMRES, FD (Upwind)

Based on Pressure Projection

- MAC (Harlow and Welch, 1965)
- Fractional Step Method (Chorin, 1968; Yanencko, 1971; Marchuck, 1975....)
- SIMPLE type Pressure Iteration (Caretto et al., 1972; Patanka & Spalding, 1972...)

Derived Variable Methods

Vorticity-Velocity (Fasel, 1976; Dennis et al., 1979; Hafez et al., 1988

Stream function-vorticity

- Spatial discretization scheme:

  Finite Difference

  Finite Volume

Governing Equations

- Incompressible Navier-Stokes Equations

$$\frac{\partial u_i}{\partial x_i} = 0$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + s$$

Where s = source term

$$\tau_{ij} = 2\upsilon S_{ij} - R_{ij} = \upsilon\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - R_{ij}$$

Eddy viscosity model for R_{ij}

$$R_{ij} = \frac{1}{3} R_{kk}\delta_{ij} - 2\nu_t S_{ij} \qquad \tau_{ij} = 2(\upsilon + \upsilon_t)S_{ij} = 2\upsilon_T S_{ij}$$

GOVERNING EQUATIONS

⊙ Coordinate Transformation

$$\xi = \xi(x,y,z,t),\ \eta = \eta(x,y,z,t),\ \zeta = \zeta(x,y,z,t)$$

- Jacobian of the transformation

$$J = \det\frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} = \begin{vmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{vmatrix}$$

- Metric terms are

$$\begin{pmatrix} \xi_x \\ \xi_y \\ \xi_z \end{pmatrix} = \frac{1}{J'}\begin{pmatrix} y_\eta z_\zeta - y_\zeta z_\eta \\ x_\zeta z_\eta - x_\eta z_\zeta \\ x_\eta y_\zeta - x_\zeta y_\eta \end{pmatrix}$$

etc...

GOVERNING EQUATIONS

⊙ Governing Equations

$$\frac{\partial}{\partial t}\hat{u} = -\frac{\partial}{\partial \xi}(\hat{e}-\hat{e}_v) - \frac{\partial}{\partial \eta}(\hat{f}-\hat{f}_v) - \frac{\partial}{\partial \zeta}(\hat{g}-\hat{g}_v) = -\hat{r}$$

$$\frac{\partial}{\partial \xi}\left(\frac{U-\xi_t}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{V-\eta_t}{J}\right) + \frac{\partial}{\partial \zeta}\left(\frac{W-\zeta_t}{J}\right) = 0$$

where

$$\hat{u} = \frac{1}{J}\begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\hat{e} = \frac{1}{J}\begin{bmatrix} \xi_x p + uU \\ \xi_y p + vU \\ \xi_z p + wU \end{bmatrix}$$

$$U = \xi_t + \xi_x u + \xi_y v + \xi_z w \quad \text{etc...}$$

GOVERNING EQUATIONS

⊙ **Viscous Terms**

$$\frac{\partial}{\partial x_j}\tau_{ij} = \frac{\partial}{\partial x_j}\nu S_{ij}$$
$$= \frac{\partial}{\partial x}\nu \begin{bmatrix} u_x + u_x \\ v_x + u_y \\ w_x + u_z \end{bmatrix} + \frac{\partial}{\partial y}\nu \begin{bmatrix} u_y + v_x \\ v_y + v_y \\ w_y + v_z \end{bmatrix} + \frac{\partial}{\partial z}\nu \begin{bmatrix} u_z + w_x \\ v_z + w_y \\ w_z + w_z \end{bmatrix}$$

For constant ν, the second terms in each bracket sums up to be zero

GOVERNING EQUATIONS

⊙ **Viscous Terms (cont'd)**

- For variable viscosity

$$\hat{e}_v = \frac{\nu}{J}\nabla\xi \cdot (\nabla\xi\frac{\partial}{\partial\xi} + \nabla\eta\frac{\partial}{\partial\eta} + \nabla\zeta\frac{\partial}{\partial\zeta}) \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$
$$+ \frac{\nu}{J}(\xi_x\frac{\partial u}{\partial\xi_i} + \xi_y\frac{\partial v}{\partial\xi_i} + \xi_z\frac{\partial w}{\partial\xi_i}) \begin{bmatrix} \frac{\partial}{\partial x}\xi_i \\ \frac{\partial}{\partial y}\xi_i \\ \frac{\partial}{\partial z}\xi_i \end{bmatrix}$$

- For constant viscosity in orthogonal coordinates

$$\hat{e}_v = (\frac{\nu}{J})({\xi_x}^2 + {\xi_y}^2 + {\xi_z}^2) \begin{bmatrix} u_\xi \\ v_\xi \\ w_\xi \end{bmatrix}$$

etc..

Artificial (or Pseudo-) Compressibility Method

- Physical characteristics
- Governing Equations in generalized Coordinates
- Steady-State Formulation
  ADI Scheme
  LU-SGS Scheme
  Numerical Dissipation
  Artificial compressibility Parameter
  Boundary Conditions
  Geometry Effects, etc.
- Time-Accurate Formulation

Artificial (or Pseudo-) Compressibility Method

- Artificial Compressibility Formulation
  Continuity equation is modified to

$$\frac{1}{\beta}\frac{\partial p}{\partial t}+\frac{\partial u_i}{\partial x_i}=0$$

- Introduces hyperbolic behavior into pressure field.
  Speed of pressure wave depends on the artificial compressibility parameter, β.
- The equations are to be marched in a time like fashion until the divergence of velocity converges to zero.
  $\Rightarrow$ Relaxes incompressibility requirement.
- Time variable during this process does not represent physical time step.

STEADY-STATE FORMULATION

⊙ Artificial Compressibility Relation

$$\frac{1}{\beta}\frac{\partial p}{\partial \tau} + \frac{\partial u_i}{\partial \xi_i} = 0$$

- The equations are to be marched in a time like fashion until the divergence of velocity converges to zero.
- The time variable for this process no longer represents physical time.
- In the momentum equations t is replaced with τ, which can be thought of as a pseudo-time or iteration parameter.

STEADY-STATE FORMULATION

⊙ **Governing Equations**

$$\frac{\partial}{\partial \tau}\hat{D} = -\frac{\partial}{\partial \xi}(\hat{E} - \hat{E}_v) - \frac{\partial}{\partial \eta}(\hat{F} - \hat{F}_v) - \frac{\partial}{\partial \zeta}(\hat{G} - \hat{G}_v) = -\hat{R}$$

where

$$\hat{D} = \frac{D}{J} = \frac{1}{J}\begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix}$$

$$\hat{E} = \begin{bmatrix} \beta(U - \xi_t)/J \\ \hat{e} \end{bmatrix} = \frac{1}{J}\begin{bmatrix} \beta(U - \xi_t) \\ \xi_x p + uU \\ \xi_y p + vU \\ \xi_z p + wU \end{bmatrix}$$

$$\hat{E}_v = \begin{bmatrix} 0 \\ \hat{e}_v \end{bmatrix}$$

NUMERICAL ALGORITHM

⊙ **Time advancing**

- Linearization

$$\hat{E}^{n+1} = \hat{E}^n + \hat{A}^n(D^{n+1} - D^n) + O(\Delta\tau^2)$$

where the Jacobian matrices are

$$\hat{A}_i = \frac{\partial \hat{E}_i}{\partial D} = \frac{1}{J}\begin{bmatrix} 0 & L_1\beta & L_2\beta & L_3\beta \\ L_1 & Q + L_1 u & L_2 u & L_3 u \\ L_2 & L_1 v & Q + L_2 v & L_3 v \\ L_3 & L_1 w & L_2 w & Q + L_3 w \end{bmatrix}$$

$$Q = L_0 + L_1 u + L_2 v + L_3 w$$

$$L_0 = (\xi_i)_t,\ L_1 = (\xi_i)_x,\ L_2 = (\xi_i)_y,\ L_3 = (\xi_i)_z$$

$$\xi_i = (\xi, \eta, \text{ or } \zeta) \text{ for } (\hat{A}, \hat{B}, \text{ or } \hat{C})$$

NUMERICAL ALGORITHM

⊙ **Time advancing**

- Trapezoidal rule

$$\hat{D}^{n+1} = \hat{D}^n + \frac{\Delta\tau}{2}\left[\left(\frac{\partial \hat{D}}{\partial \tau}\right)^n + \left(\frac{\partial \hat{D}}{\partial \tau}\right)^{n+1}\right] + O(\Delta\tau^3)$$

Then the governing equation becomes

$$D^{n+1} + \frac{\Delta\tau}{2}J\left[\delta_\xi(\hat{E} - \hat{E}_v)^{n+1} + \delta_\eta(\hat{F} - \hat{F}_v)^{n+1} + \delta_\zeta(\hat{G} - \hat{G}_v)^{n+1}\right]$$
$$= D^n - \frac{\Delta\tau}{2}J\left[\delta_\xi(\hat{E} - \hat{E}_v)^n + \delta_\eta(\hat{F} - \hat{F}_v)^n + \delta_\zeta(\hat{G} - \hat{G}_v)^n\right]$$

NUMERICAL ALGORITHM

⊙ **Governing equation in delta form**

$$\left\{I + \frac{h}{2}J^{n+1}\left[\delta_\xi(\hat{A}^n - \Gamma_1) + \delta_\eta(\hat{B}^n - \Gamma_2) + \delta_\zeta(\hat{C}^n - \Gamma_3)\right]\right\}(D^{n+1} - D^n)$$
$$= -\Delta\tau J^{n+1}\left[\delta_\xi(\hat{E} - \hat{E}_v)^n + \delta_\eta(\hat{F} - \hat{F}_v)^n + \delta_\zeta(\hat{G} - \hat{G}_v)^n\right]$$
$$+ \left(\frac{J^{n+1}}{J^n} - 1\right)D^n \ = \ RHS$$

⊙ **Approximate Factorization**

$$L_\xi L_\eta L_\zeta(D^{n+1} - D^n) = RHS$$

where

$$L_\xi = \left[I + \frac{\Delta\tau}{2}J^{n+1}\delta_\xi(\hat{A}^n - \gamma_1)\right]$$
$$L_\eta = \left[I + \frac{\Delta\tau}{2}J^{n+1}\delta_\eta(\hat{B}^n - \gamma_2)\right]$$
$$L_\zeta = \left[I + \frac{\Delta\tau}{2}J^{n+1}\delta_\zeta(\hat{C}^n - \gamma_3)\right]$$

NUMERICAL ALGORITHM

⊙ **Approximate Factorization**

- Second-order central differencing :
  - Block tridiagonal inversion

$$(L_\eta)\Delta\bar{D} = RHS$$
$$(L_\xi)\Delta\bar{\bar{D}} = \Delta\bar{D}$$
$$(L_\zeta)\Delta D^{n+1} = \Delta\bar{\bar{D}}.$$

  - Diagonal form is also available

- Factorization error : cross-product terms

$$h^2\left[\delta_\xi A\delta_\eta B + \delta_\eta B\delta_\zeta C + \delta_\zeta C\delta_\xi A\right]\Delta D + O(h^3)$$

where

$$A = \hat{A}^n - \gamma_1, \quad B = \hat{B}^n - \gamma_2, \quad C = \hat{C}^n - \gamma_3, \quad h = \frac{\Delta\tau}{2}J^{n+1}$$

NUMERICAL DISSIPATION (SMOOTHING)

⊙ Higher order smoothing terms are added for stability

$$L_\xi L_\eta L_\zeta (D^{n+1} - D^n) = \text{RHS } - \epsilon_e[(\nabla_\xi \Delta_\xi)^2 + (\nabla_\eta \Delta_\eta)^2 + (\nabla_\zeta \Delta_\zeta)^2] D^n$$

where

$$L_\xi = \left[I + \frac{\Delta\tau}{2} J^{n+1} \delta_\xi (\hat{A}^n - \gamma_1) + \epsilon_i \nabla_\xi \Delta_\xi \right]$$
$$L_\eta = \left[I + \frac{\Delta\tau}{2} J^{n+1} \delta_\eta (\hat{B}^n - \gamma_2) + \epsilon_i \nabla_\eta \Delta_\eta \right]$$
$$L_\zeta = \left[I + \frac{\Delta\tau}{2} J^{n+1} \delta_\zeta (\hat{C}^n - \gamma_3) + \epsilon_i \nabla_\zeta \Delta_\zeta \right]$$

IMPLICIT vs EXPLICIT DISSIPATION

⊙ 1-D continuity equation : $\frac{1}{\beta}\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} = 0$

⊙ AF algorithm in delta form
(assuming divergence free velocity at time level n)

$$1 - \epsilon_i \nabla_\xi \Delta_\xi (p^{n+1} - p^n) = -\epsilon_e (\nabla_\xi \Delta_\xi)^2 p^n$$

⊙ Discrete Fourier expansion of p

$$p = \sum_n \hat{p}(k) e^{ik\xi}$$

where

$$k = \frac{2\pi}{N\Delta\xi} n = wave\, number$$
$$n = -N/2, ...0, 1, ..(N/2 - 1)$$
$$N = number\, of\, mesh\, points$$

IMPLICIT vs EXPLICIT DISSIPATION (cont'd)

⊙ This leads to $[1 - \epsilon_i k'](\hat{p}^{n+1} - \hat{p}^n) = -\epsilon_e (k')^2 \hat{p}^n$

where $k' = -2 + 2cos(k)$ and $(k')^2 = 6 - 8cos(k) + 2cos(2k)$

⊙ Amplification factor

$$\sigma = \frac{\hat{p}^{n+1}}{\hat{p}^n} = \frac{[1 - \epsilon_i k' - \epsilon_e (k')^2]}{[1 - \epsilon_i k']}$$

⊙ To damp out numerical fluctuation

$$|\sigma| < 1$$

$$2\epsilon_e \leq \epsilon_i$$

EXPLICIT DISSIPATION ON ACCURACY

⊙ 1-D continuity in AF form

$$[1 - \epsilon_i \nabla_x \Delta_x](p^{n+1} - p^n) = -\beta h \delta_x u - \epsilon_e (\nabla_x \Delta_x)^2 p^n$$

⊙ At steady state, this leads to

$$\delta_x u \longrightarrow -\frac{\epsilon_e (\nabla_x \Delta_x)^2 p^n}{\beta h}$$

- Mesh refinement usually does not help
- This error may be reduced (in a crude way) by

$$\epsilon_e = (\epsilon_e)_o e^{-\alpha(t - t_o)}$$

where $(\epsilon_e)_o$ is the value from $t = 0$ to $t = t_o$

Pressure Wave Propagation vs Viscous Effect

⊙ Locally linearized momentum equation

$$\frac{\partial^2 p}{\partial t^2} + 2u\frac{\partial^2 p}{\partial t \partial x} - \beta\frac{\partial^2 p}{\partial x^2} = \beta\frac{\partial \tau_w}{\partial x}$$
$$\frac{\partial^2 u}{\partial t^2} + 2u\frac{\partial^2 u}{\partial t \partial x} - \beta\frac{\partial^2 u}{\partial x^2} = -\frac{\partial \tau_w}{\partial t}$$

These may be written as

$$\left[\frac{\partial}{\partial t} + (u+c)\frac{\partial}{\partial x}\right]\left[\frac{\partial}{\partial t} + (u-c)\frac{\partial}{\partial x}\right]\begin{pmatrix} p \\ u \end{pmatrix} = \begin{pmatrix} \beta\partial\tau_w/\partial x \\ -\partial\tau_w/\partial t \end{pmatrix}$$

⊙ Characteristic equation for linear waves (without shear stress terms)

$$\left[\frac{\partial}{\partial t} + (u+c)\frac{\partial}{\partial x}\right]\begin{pmatrix} p^+ \\ u^+ \end{pmatrix} = 0$$
$$\left[\frac{\partial}{\partial t} + (u-c)\frac{\partial}{\partial x}\right]\begin{pmatrix} p^- \\ u^- \end{pmatrix} = 0$$

Pressure Wave Propagation vs Viscous Effect

⊙ One-dimensional characteristic equation (without viscous term)

$$\frac{\partial u}{\partial t} + \frac{1}{(u \pm c)}\frac{\partial p}{\partial t} + \left[\frac{\partial u}{\partial x} + \frac{1}{(u \pm c)}\frac{\partial p}{\partial x}\right](u \pm c) = 0$$

⊙ Pseudo speed of sound

$$c = \sqrt{u^2 + \beta}$$

⊙ Pseudo Mach number

$$M = \frac{u}{c} = \frac{u}{\sqrt{u^2+\beta}} < 1$$

$\Rightarrow$ M is always less than 1 for all $\beta > 0$

Criterion for β

⊙ Two Time Scale

- Time required for upstream propagating wave to travel a distance, x_L

$$\tau_L = \frac{x_L}{c-u}$$

- Time scale for viscous effect to spread a distance, δ

$$\tau_\delta = \frac{\mathrm{Re}}{4}\left(\frac{\delta}{x_{ref}}\right)^2$$

⊙ Decoupling Requirement

$$\tau_\delta \gg \tau_L$$

⊙ Lower Bound of β

$$\beta \gg \left[1 + \frac{4}{Re}\left(\frac{x_{ref}}{\delta}\right)^2\left(\frac{x_L}{x_{ref}}\right)\right]^2 - 1$$

CRITERIA FOR β : LOWER BOUND

⊙ **Physical Requirement for Incompressibility**

The pressure wave must propagate much faster than the spreading of viscous effect :

$$\tau_L << \tau_\delta$$

τ_L = time for pressure field to map the flow field
τ_δ = time for viscous effect to spread

⊙ **Lower Bound of β**

$$\beta >> \left[1 + \frac{4}{Re}\left(\frac{x_{ref}}{x_\delta}\right)^2\left(\frac{x_L}{x_{ref}}\right)\right]^2 - 1 \qquad \text{(Laminar)}$$

$$\beta >> \left[1 + \frac{1}{Re_t}\left(\frac{x_{ref}}{x_\delta}\right)^2\left(\frac{x_L}{x_{ref}}\right)\right]^2 - 1 \qquad \text{(Turbulent)}$$

CRITERIA FOR β : UPPER BOUND

⊙ **Governing Equation in 'Delta Form '**

$$\Big(I + \frac{h}{2}J^{n+1}[\delta_\xi(A_1{}^n - \Gamma_1) + \delta_\eta(A_2{}^n - \Gamma_2)$$
$$+\delta_\zeta(A_3{}^n - \Gamma_3)]\Big)(Q^{n+1} - Q^n) = RHS$$

⊙ **Approximately Factored Form**

$$\left[I + \frac{h}{2}J^{n+1}\delta_\xi(A_1{}^n - \Gamma_1)\right] \cdot \left[I + \frac{h}{2}J^{n+1}\delta_\eta(A_2{}^n - \Gamma_2)\right]$$
$$\cdot\left[I + \frac{h}{2}J^{n+1}\delta_\zeta(A_3{}^n - \Gamma_3)\right]\Delta Q = RHS$$

⇒ Higher order cross differencing terms are added.

CRITERIA FOR β : UPPER BOUND

⊙ **Added terms must be kept smaller than the original terms**

$$O\left(\frac{h}{2}J\delta_\xi A_1\right) > O\left((\frac{h}{2}J)^2\delta_\xi A_1\delta_\eta A_2\right)$$

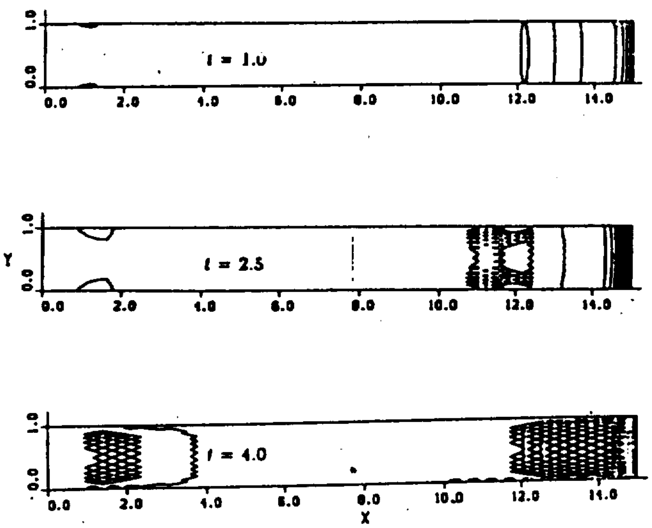
⊙ **Upper Bound of β**

$$\beta h < O(1)$$

Artificial Pressure Wave in Channel

Re=1000, β=0.1

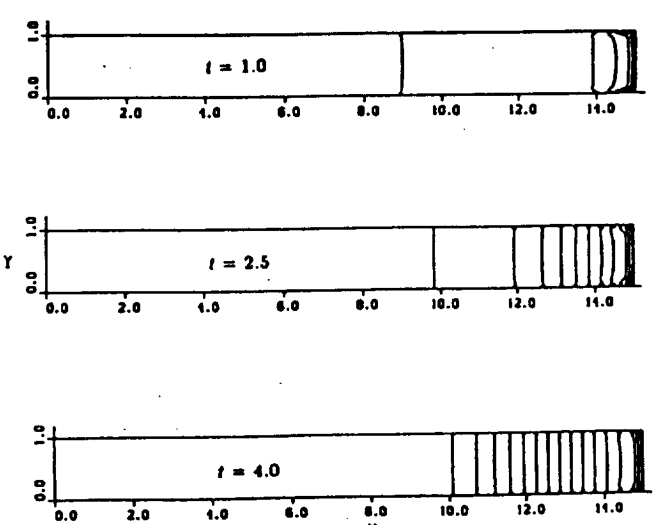
(Recommended Range: 0.12< β < 10.0)



Artificial Pressure Wave in Channel

Re=1000, β=5.0

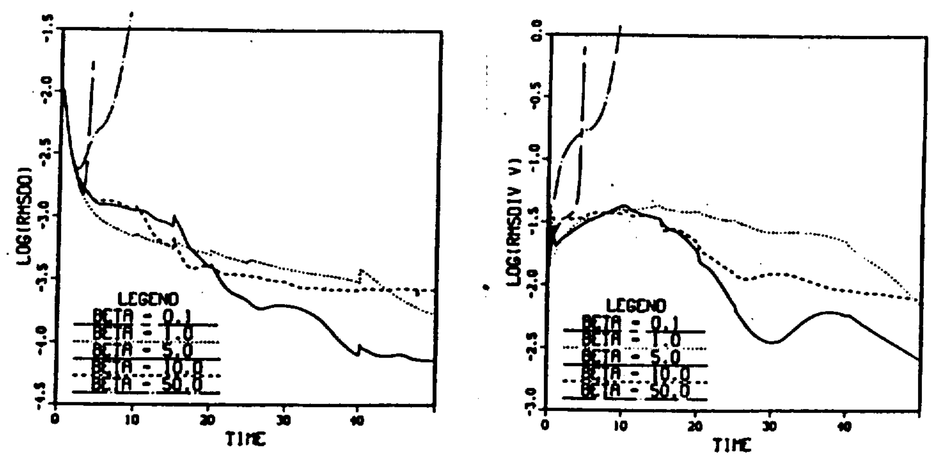
(Recommended Range: 0.12< β < 10.0)



<u>Convergence History for Channel Flow</u>

Re=1000, Channel Length = 15.0

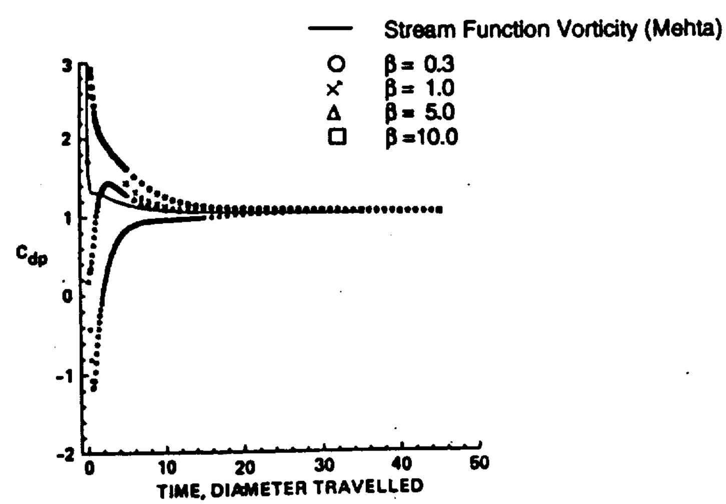
(Recommended Range: 0.12< β < 10.0)



<u>Artificial Compressibility on Convergence</u>

Impulsively started circular cylinder to steady-state at Re=40

(Recommended Range for β : 0.1< β < 10.0)



NUMERICAL ALGORITHM

⊙ **LU-SGS Scheme**

(Implicit Lower-Upper Symmetric-Gauss-Seidel scheme)

- Start from an unfactored implicit scheme

$$\left\{I + \frac{h}{2}\left[\delta_\xi \hat{A} + \delta_\eta \hat{B} + \delta_\zeta \hat{C}\right]\right\}(D^{n+1} - D^n)$$

$$= -\Delta t\left[\delta_\xi(\hat{E} - \hat{E}_v) + \delta_\eta(\hat{F} - \hat{F}_v) + \delta_\zeta(\hat{G} - \hat{G}_v)\right]$$

- LU-SGS implicit factorization scheme

$$L_l {L_d}^{-1} L_u = RHS$$

where

$$L_l = I + \frac{h}{2}(\delta^-{}_\xi \hat{A}^+ + \delta^-{}_\eta \hat{B}^+ + \delta^-{}_\zeta \hat{C}^+ - \hat{A}^- - \hat{B}^- - \hat{C}^-)$$

$$L_d = I + \frac{h}{2}(\hat{A}^+ - \hat{A}^+ + \hat{B}^+ - \hat{B}^- + \hat{C}^+ - \hat{C}^-)$$

$$L_u = I + \frac{h}{2}(\delta^+{}_\xi \hat{A}^- + \delta^+{}_\eta \hat{B}^- + \delta^+{}_\zeta \hat{C}^- + \hat{A}^+ + \hat{B}^+ + \hat{C}^+)$$

Downstream Boundary Conditions

⊙ Velocities

- Extrapolate to update $(\hat{u}^n)_{L_{max}}$
- Calculate mass flux out

$$\dot{m}_{out} = \int_{Ae} \hat{u}^n \cdot d\hat{a}$$

- Then, obtain weighted velocities to conserve mass

$$(\hat{u}^n)_{L_{max}} = \frac{\dot{m}_{in}}{\dot{m}_{out}} (\hat{u}^n)_{L_{max}}$$

Downstream Boundary Conditions (cont'd)

⊙ Pressure

- ζ-momentum equation at $L = L_{max} - 1$ and $t = t^{\hat{n}}$

$$[\partial_\tau \hat{w} + \partial_\xi \hat{e}_1 + \partial_\eta \hat{e}_2]^n + [\partial_\zeta \hat{e}_3]^{\hat{n}} = 0$$

where $\hat{e}_i = \frac{1}{J}\left[wU_i + (\xi_i)_z p\right] - \frac{\nu}{J}(\nabla\xi_i \cdot \nabla\xi_j)\frac{\partial w}{\partial \xi_j}$

- Pressure corresponding to mass weighted velocities

$$p^{\hat{n}} = p^n - \frac{1}{\zeta_z}\left[(wU_3)^{\hat{n}} - (wU_3)^n\right] + \frac{\nu}{\zeta_z}(\nabla\zeta \cdot \nabla\zeta)\left[\left(\frac{\partial w}{\partial \zeta}\right)^{\hat{n}} - \left(\frac{\partial w}{\partial \zeta}\right)^n\right]$$

(Variation in velocities normal to ζ-direction is assumed to be negligible)

Downstream Boundary Conditions (cont'd)

⊙ Pressure (cont'd)

- **Progressive correction method**

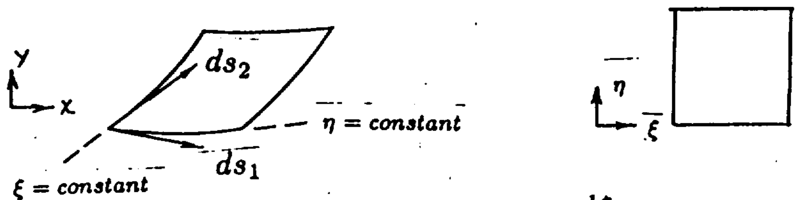
$$\tilde{I}_p = \int_{exit} p^{\tilde{n}} da$$

$$I_p = \int_{exit} p^{n} da$$

A momentum-weighted pressure is then formed as

$$p^{\hat{n}} = \left(\frac{\tilde{I}_p}{I_p}\right) p^{n}$$

Skewness and Metrics



$$ds_1 \cdot ds_1 = ({\eta_x}^2 + {\eta_y}^2)(\frac{d\xi}{J})^2$$

$$ds_2 \cdot ds_2 = ({\xi_x}^2 + {\xi_y}^2)(\frac{d\eta}{J})^2$$

$$ds_1 \cdot ds_2 = -(\xi_x \eta_x + \xi_y \eta_y)\frac{d\xi d\eta}{J^2})$$

$$\frac{ds_1 \cdot ds_2}{({ds_1}^2 {ds_2}^2)^{1/2}} = cos\theta$$

Skewness and Viscous Terms

⊙ AF Algorithm for Incomp Navier-Stokes Eqs

$$\left[I + \frac{h}{2} J\delta_\xi(A_1 - e_{v1} + \epsilon_i \nabla_\xi \Delta_\xi)\right] [...] [...] (Q^{n+1} - Q^n) = RHS$$

$$RHS = -\Delta\tau J\left[\delta_\xi(E_1 - E_{v1})...\right]$$

⊙ Viscous Term

$$E_{v1} = \frac{\nu}{J}\left[\nabla\xi_i \cdot \nabla\xi_j I_m \delta_{\xi j} Q + ...\right]$$

For orthogonal grid, (and for LHS)

$$e_{v1} = \frac{\nu}{J}\left[\nabla\xi_i \cdot \nabla\xi_i I_m \delta_{\xi i} Q + ...\right]$$

Implicit Scheme

⊙ Linearize the right-hand side, using first order upwind difference and approximate Jacobians, resulting in a system of equations of the form

$$\mathcal{B}[T,0,...,0,U,0,...,0,X,Y,Z,0,...,0,V,0,...,0,W]\Delta D = R$$

⊙ Solve iteratively with line relaxation

$$\begin{aligned}\mathcal{B}[X,Y,Z]\Delta D^{l+1} = R - T\Delta D^{l+1}_{i,j,k-1} - U\Delta D^{l+1}_{i,j-1,k} \\ - V\Delta D^{l}_{i,j+1,k} - W\Delta D^{l}_{i,j,k+1}\end{aligned}$$

⊙ Solve efficiently by

1) First form and store T, U, V, W, X, Y, Z, and R for all grid points
2) Perform LU decomposition of left-hand side, overwriting X, Y, and Z
3) Iteratively form right-hand side and solve

Boundary Conditions

⊙ Inflow and Outflow boundaries based on Method of Characteristics

⊙ Inflow Boundary

- One upstream traveling characteristic
- Three flow components specifed, either
  All velocity components, or
  Total pressure and flow direction

⊙ Outflow Boundary

- Three downstream traveling characteristics
- Static pressure

Upwind Differencing I

⊙ Because artificial compressibility is used, convective terms form a hyperbolic system of equations

- Upwind differencing used to follow propagation of artificial characteristic waves
- Provides dissipation to suppress oscillations caused by the non-linear convective terms
- Produces a more diagonally dominant system of equations than central differencing
- More expensive to compute, but convergence improves considerably

⊙ Flux-difference splitting with Roe averaging is used

⊙ Implemented as 3rd or 5th order accurate throughout all the interior grid points - no limiting

Upwind Differencing II

$$\frac{\partial \hat{E}}{\partial \xi} \approx \frac{[\tilde{E}_{i+1/2} - \tilde{E}_{i-1/2}]}{\Delta \xi}$$

$$\tilde{E}_{i+1/2} = \frac{1}{2}\left[\hat{E}(D_{i+1}) + \hat{E}(D_i) - \phi_{i+1/2}\right]$$

⊙ $\phi_{i+1/2}$ provides dissipaton, function of the flux difference $\Delta E^{\pm}$

$$\Delta E^{\pm}_{i+1/2} = A^{\pm}(\bar{D})(D_{i+1} - D_i)$$

⊙ Roe properties are satisfied for $\bar{D} = 0.5(D_{i+1} + D_i)$

$$A^{\pm} = X_1 \Lambda_1^{\pm} X_1^{-1}$$

$$\lambda_i^{\pm} = \frac{1}{2}\left(\lambda_i + \sqrt{\lambda_i^2 + \epsilon}\right)$$

TIME-ACCURATE FORMULATION

- Time-accurate formulation not as straight forward - no $\frac{\partial p}{\partial t}$ term
- First discretize the time term in momentum equations using second-order three-point backward-difference formula

$$\left(\frac{\partial \hat{U}}{\partial \xi}+\frac{\partial \hat{V}}{\partial \eta}+\frac{\partial \hat{W}}{\partial \zeta}\right)^{n+1}=0$$

$$\frac{3\hat{u}^{n+1}-4\hat{u}^{n}+\hat{u}^{n-1}}{2\Delta t}=-\hat{r}^{n+1}$$

- Iteratively solve these equations by introducing a pseudo-time level and artificial compressibility

$$\frac{1}{\Delta\tau}(\hat{p}^{n+1,m+1}-\hat{p}^{n+1,m})=-\beta\nabla\cdot u^{n+1,m+1}$$

$$\frac{1.5}{\Delta t}(\hat{u}^{n+1,m+1}-\hat{u}^{n+1,m})=-\hat{r}^{n+1,m+1}-\frac{3\hat{u}^{n+1,m}-4\hat{u}^{n}+\hat{u}^{n-1}}{2\Delta t}$$

Resulting Systems of Equations

- Steady-state formulation

$$\left[\frac{1}{J\Delta\tau}I+\left(\frac{\partial \hat{R}}{\partial D}\right)^{n}\right]\Delta D=-\hat{R}^{n}$$

- Time-accurate formulation

$$\left[\frac{I_{t\tau}}{J}+\left(\frac{\partial \hat{R}}{\partial D}\right)^{n+1,m}\right]\Delta D=-\hat{R}^{n+1,m}-\frac{I_m}{2\Delta t}(3\hat{D}^{n+1,m}-4\hat{D}^{n}+\hat{D}^{n-1})$$

- Same residual $\hat{R}$ for both formulations, formed using
  - Upwind differencing for convective fluxes
  - Central differencing for viscous fluxes

Summary of Time-Accurate Formulation of Artificial Compressibility Approach

- Time accuracy is achieved by subiteration
  - Discretize the time term in momentum equations using second-order three-point backward-difference formula

$$\frac{3q^{n+1} - 4q^{n} + q^{n-1}}{2\Delta t} = -(rhs)^{n+1}$$

  - Introduce a pseudo-time level and artificial compressibility,
  - Iterate the equations in pseudo-time for each time step until incompressibility condition is satisfied.

$$\frac{1}{\Delta\tau}(p^{n+1,m+1} - p^{n+1,m}) = -\beta q^{n+1,m+1}$$

$$\frac{1.5}{\Delta\tau}(q^{n+1,m+1} - q^{n+1,m}) = -(rhs)^{n+1,m+1} - \frac{3q^{n+1,m} - 4q^{n} + q^{n-1}}{2\Delta t}$$

- Code performance
  - Computing time : 50-120 ms/grid point/iteration
  - Memory usage: **Line-relaxation** 45 words/grid point
    GMRES-ILU(0) 220 words/grid point

Pressure Projection Methods

- Pressure is used as a mapping parameter to maintain incompressibility, that is, to maintain divergence free velocity field
  - ⇒ This is accomplished via derived equation, i.e. Poisson equation for pressure
- The solution is usually done in multi-step (fractional step method)
- Basic idea of this approach will be illustrated by presenting
  - MAC method
    Issues n pressure field computation
  - A form of SIMPLE method
  - A generic pressure projection method via fractional step approach
- A 3-D generalized method will be presented in detail

Marker-and-Cell (MAC) Method

Harlow and Welch (1965)

⊙ **Poisson Equation for Pressure**
(Taking divergence of momentum eqn)

$$\nabla^2 p = \frac{\partial h_i}{\partial x_i} - \frac{\partial}{\partial t}\frac{\partial u_i}{\partial x_i} = g$$

where

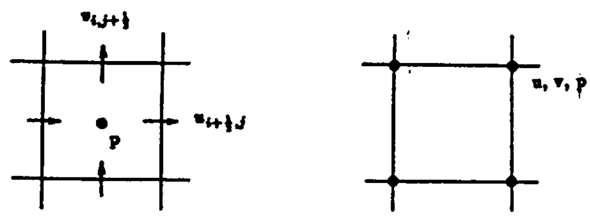
$$h_i = -\frac{\partial u_i u_j}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j}$$

⊙ **Solution Procedure**

- Momentum equations are solved for velocity
- Poisson equation is solved for pressure requiring divergence free velocity field at the next time level

MAC Method

⊙ Grid



staggered regular

⊙ Original MAC method : staggered Cartesian grid

- Conserve mass, momentum and kinetic energy
- Avoid odd-even point decoupling

⇒ In generalized coord these advantages become unclear

⊙ Poisson solver requires large computing time

MAC Method

⊙ Important to get numerical divergence free velocity field

⊙ **Poisson Solver**

Method 1 : Use an exact form of the Laplacian

$$\nabla^2 p = g'$$

Fourier transform of this

$$-k^2 \hat{p} = \hat{g}'$$

where

$$\hat{p} = \text{Fourier transform of p}$$
$$k^2 = {k_x}^2 + {k_y}^2 + {k_z}^2$$
$$k_x, .. = \frac{2\pi}{N\Delta} n = \text{wave number in x} - \text{direction}$$
$$g' = \text{finite difference approximation to} \quad g$$
$$n = -N/2, ...0, 1, ..(N/2-1)$$

MAC Method

⊙ **Poisson Solver**

Method 2 : Use difference form of second derivative

$$(\frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2} + \frac{\delta^2}{\delta z^2})p = g'$$

Fourier transform of this

$$-(\tilde{k}_i)^2 \hat{p} = \hat{g}'$$

where

$$(\tilde{k}_i)^2 = \text{Fourier transform of } \frac{\delta^2}{\delta x_i \delta x_i}$$

(i.e. for a five point central differencing)

$$(\tilde{k}_i)^2 = \frac{1}{6\Delta^2}[15 - 16cos(\Delta k_i) + cos(2\Delta k_i)]$$

MAC Method

⊙ **Poisson Solver**

Method 3 : Use divergence-gradient operator

Finite difference form of the governing equations are

$$\frac{\delta u_i}{\delta t} - h_i' = -\frac{\delta p}{\delta x_i} = Gp$$

$$\frac{\delta u_i}{\delta x_i} = Du_i = 0$$

where

$$h_i' = -\frac{\delta u_i u_j}{\delta x_j} + \frac{\delta \tau_{ij}}{\delta x_j}$$

MAC Method

⊙ **Poisson Solver**

Method 3 : Use divergence-gradient operator (cont'd)

Applying the divergence operator, D,

$$DGp = -\frac{\delta Du_i}{\delta t} + Dh'_i = g'_i$$

Fourier transform of this : $-k'_i k'_i \hat{p} = \hat{g}'$
where $(k'_i)^2$ = Fourier transform of DG operator
(i.e. for a five point central differencing)

$$(k_i')^2 = \frac{1}{72\Delta^2}\big[65 - 16cos(\Delta k_i) - 64cos(2\Delta k_i) + 16cos(2\Delta k_i) - cos(4\Delta k_i)\big]$$

MAC Method

⊙ **Poisson Solver**

- Fourier transform of momentum equation

$$\frac{\delta \hat{u}_i}{\delta t} - \hat{h}'_i = -ik'_i\hat{p}$$

To satisfy the continuity equation in grid space

$$\frac{\delta}{\delta t}(k'_i\hat{u}_i) = 0$$

MAC Method

⊙ Poisson Solver

- Substituting $\hat{p}$ fromm the three methods into the above

  For Method 1:

$$\frac{\delta}{\delta t}(k_i'\hat{u}_i) = \big(\hat{h}_i' - \frac{k_i'k_j'}{k^2}\hat{h}_j'\big)k_i' \neq 0$$

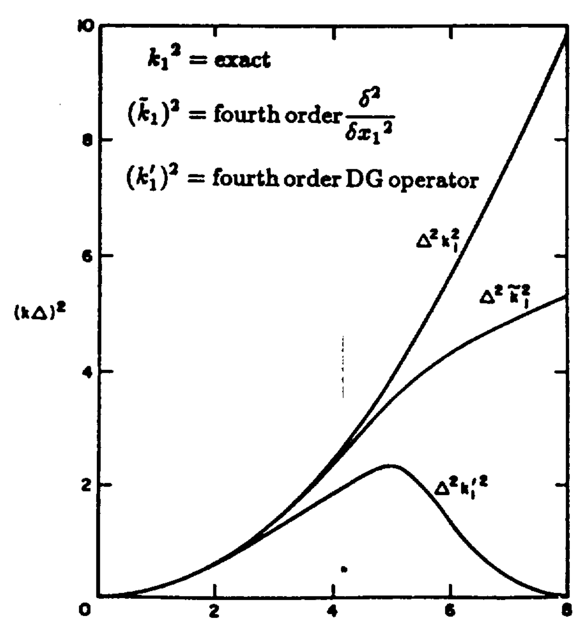
  For Method 2:

$$\frac{\delta}{\delta t}(k_i'\hat{u}_i) = \big(\hat{h}_i' - \frac{k_i'k_j'}{\tilde{k}^2}\hat{h}_j'\big)k_i' \neq 0$$

  For Method 3:

$$\frac{\delta}{\delta t}(k_i'\hat{u}_i) = \big(\hat{h}_i' - \frac{k_i'k_j'}{k'^2}\hat{h}_j'\big)k_i' = 0$$

Comparison of Laplacian Operator
(16 equally spaced 1-D mesh)



Semi-Implicit Method for Pressure-Linked Equations

Caretto, et al. (1972), Patankar and Spalding (1972)

⊙ Computing correct pressure for a divergence free velocity field in each step (i.e. Poisson solver for pressure) slows down the overall computational efficiency

⊙ For a steady-state solution, the correct pressure field is desired only when the solution is converged

⊙ Iteration procedure for the pressure can be simplified such that it requires only a few iteration at each time step.

⇒ The most well known of this approach is the SIMPLE method

SIMPLE Method

⊙ **Solution Procedure**

- Begin with a guessed pressure p^*
  (Usually p^n at the beginning of the cycle)
- Solve momentum equation to get an intermediate velocity $u_i{}^*$

$$u_i{}^* - u_i{}^n = \Delta t\left[fcn(u^n, v^n, u^*, v^*) - \frac{\delta p^*}{\delta x_i}\right]$$

- Corrected pressure and velocity are obtained by

$$p = p^* + p'$$

$$u_i = u_i{}^* + u_i{}'$$

SIMPLE Method

⊙ Solution Procedure (cont'd)

- Relation between p' and u'
  - First, linearize the momentum equations
  - Then, drop all terms involving neighboring velocities

$$u'_i = (\text{fcn differencing scheme})\frac{\delta p'}{\delta x_i}$$

- Substituting these into the continuity equation, a pressure correction equation is obtained

SIMPLE Method

⊙ Salient Features

- This procedure in essence results in a simpliified Poisson equation for pressure, which can be solved iteratively line-by-line.
- The unique feature of this method comes from the simple way of estimating the velocity correction u_i'.
- This feature simplifies the computation but introduces empiricism into the method.
- Despite its empiricism, many computations have been done successfully using various forms of this method.

Fractional Step Method

Chorin (1968), Yanenko (1971), Marchuk (1975)

- ⊙ Time evolution is approximated by several steps.
- ⊙ Various operator splitting can be adopted by treating the momentum equation as a combination of convection, pressure, and viscous terms.
- ⊙ The common application of this method is done by two steps.

  (1) Solve for an intermediate velocity field using a simplified momentum equation.

  (2) Compute pressure which maps auxiliary velocity onto a divergence-free velocity field.

Fractional Step Method

⊙ Solution Procedure (an example)

- Step 1: Calculate intermediate velocity, $\hat{u}_i$
  (i.e. by a second order Adams-Bashforth method)

$$\frac{\hat{u}_i - u_i{}^n}{\Delta t} = \frac{1}{2}(3H_i{}^n - H_i{}^{n-1}) - \frac{\delta p^n}{\delta x_i} + \frac{1}{2}\frac{1}{Re}\nabla^2(\hat{u}_i + u_i{}^n)$$

where

$$H_i = -\frac{\delta}{\delta x_j} u_i u_j$$

Fractional Step Method

⊙ Solution Procedure (cont'd)

- Step 2: Solve for the pressure correction

$$\frac{u_i{}^{n+1} - \hat{u}_i}{\Delta t} = -\frac{1}{2}\frac{\delta}{\delta x_i}(\phi^{n+1} - \phi^n)$$

where

$$p^n = \phi^n - \frac{\Delta t}{2Re}\nabla^2\phi^n$$

This combined with continuity equation results in Poisson equation for pressure correction.

$$\nabla^2(\phi^{n+1} - \phi^n) = \frac{2}{\Delta t}\frac{\delta}{\delta x_i}\hat{u}_i$$

Fractional Step Method

⊙ Solution Procedure (cont'd)

- New pressure and velocities are calculated as follows:

$$p^{n+1} = p^n + (\phi^{n+1} - \phi^n) - \frac{\Delta t}{2Re}\nabla^2(\phi^{n+1} - \phi^n)$$
$$u_i^{n+1} = \hat{u}_i - \frac{\Delta t}{2}\frac{\delta}{\delta x_i}(\phi^{n+1} - \phi^n)$$

Fractional Step Method

⊙ Salient Features

- Need special care for intermediate boundary conditions (Orszag et al., 1986)
- As other pressure based methods, efficiency depends on the Poisson solver.
- A multigrid acceleration is one possible avenue to enhance the computational efficiency.

Pressure Projection in Generalized Coordinates

- Approach in generalized coordinates
  - Finite volume discretization
  - Accurate treatment of geometric quantities
  - Dependent variables - pressure and volume fluxes
    Mass and momentum conservation
  - Implicit time integration
  - Fractional step procedure
    Solve auxiliary velocity field first,
    then enforce incompressibility condition by solving a Poisson equation for pressure.

- INS3D-FS Code performance
  - Computing time : 80 ms/grid point/iteration
  - Memory usage: 70 words/grid point

Pressure Projection Method

- Fractional-step
  - Solve for the auxiliary velocity field, using implicit predictor step:

$$\frac{1}{\Delta t}(u_i^* - u_i^*) = -\nabla p^* + h(u_i^*)$$

  - The velocity field at time level (n+1) is obtained by using a correction step:

$$\frac{2}{\Delta t}(u_i^{n+1} - u_i^*) = -\nabla p^{n+1} + h(u_i^{n+1}) - \nabla p^* + h(u_i^*)$$

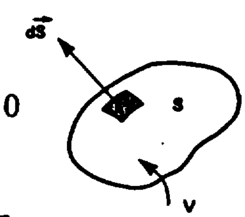
  - The incompressibility condition is enforced by using a Poisson equation for pressure ($p' = p^{n+1} - p^*$)

$$\nabla^2 p' = \frac{2}{\Delta t}\nabla \cdot u^*$$

FORMULATION - I

- Mass conservation

$$\frac{\partial V}{\partial t} + \oint_S d\vec{S} \cdot (\vec{u} - \vec{v}) = 0$$



- Momentum conservation

$$\frac{\partial}{\partial t} \int_V \vec{u}\, dV = \oint_S d\vec{S} \cdot \bar{T}$$

$$\bar{T} = -(\vec{u} - \vec{v})\vec{u} - P\,\bar{I} + \nu\left(\nabla\vec{u} + (\nabla\vec{u})^T\right)$$

FORMULATION - II

- Conservation of volume for time-varying cell

$$\frac{\partial V}{\partial t} - \oint_S d\vec{S} \cdot \vec{v} = 0$$

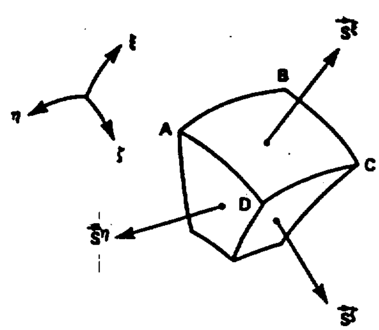
- Mass conservation

$$\oint_S d\vec{S} \cdot \vec{u} = 0$$

DISCRETIZATION – I
Geometry

- Closed cell

$$\sum_{l=faces} \vec{S}^l = 0$$



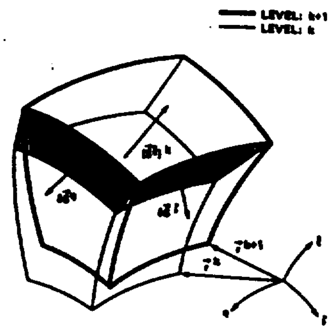
- No gaps or overlapping cells

$$\sum_{cells} V = V_{total}$$

DISCRETIZATION – II
Geometry (cont.)

- Conservation of volume for time-varying cell can be satisfied by

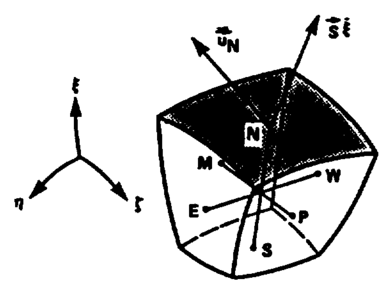
$$\mathcal{V}^{k+1} = \mathcal{V}^k + \sum_l \delta \mathcal{V}^l$$



DISCRETIZATION – III
Mass Conservation

$$(\vec{S}^\xi \cdot \vec{u})_N - (\vec{S}^\xi \cdot \vec{u})_S + (\vec{S}^\eta \cdot \vec{u})_E - (\vec{S}^\eta \cdot \vec{u})_W$$
$$+(\vec{S}^\zeta \cdot \vec{u})_M - (\vec{S}^\zeta \cdot \vec{u})_P = 0$$

$$U^\xi = \vec{S}^\xi \cdot \vec{u}$$
$$U^\eta = \vec{S}^\eta \cdot \vec{u}$$
$$U^\zeta = \vec{S}^\zeta \cdot \vec{u}$$



$$U_N^\xi - U_S^\xi + U_E^\eta - U_W^\eta + U_M^\zeta - U_P^\zeta = 0$$

DISCRETIZATION – III
Momentum Conservation

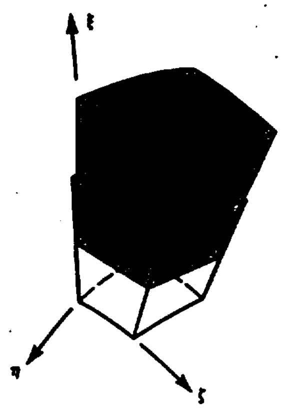
$$V\frac{\partial \vec{u}}{\partial t} = \sum_l \vec{S}^l \cdot \bar{T}^l = \vec{F}$$

U^ξ – momentum:

$$\vec{S}^\xi \cdot \left(V\frac{\partial \vec{u}}{\partial t} \right) = V\frac{\partial U^\xi}{\partial t} = \vec{S}^\xi \cdot \vec{F}$$

since

$$\vec{S}^\xi \cdot \vec{u} = U^\xi$$



DISCRETIZATION – IV
Summary of properties

- Dependent variables: U^l, P
- Staggered grid
- Conservative scheme
- Second-order in space
- First-order in time
- Implicit scheme
- Fourth-order conservative numerical dissipation

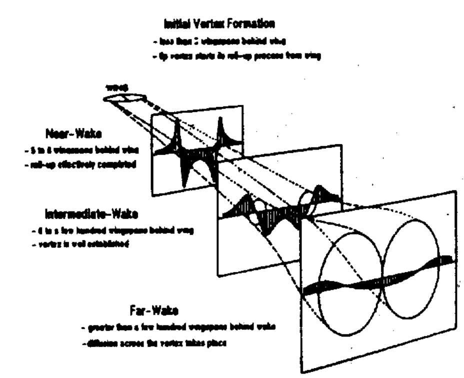
Test Problems for Code Validation

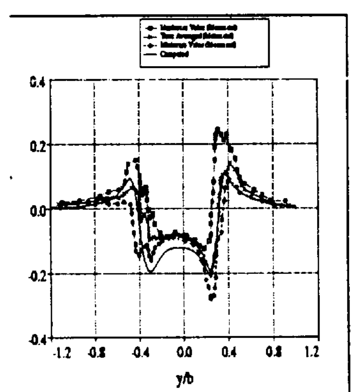
- Two problems are selected to validate INS3D codes and compare artificial compressibility method vs pressure projection method

  - Wake and wing tip vortex propagation
  - Impulsively started flat plate

Performance of any codes will depend on many factors beyond algorithm. However, it is assumed that both algorithms are reasonably well coded and optimized at a similar level

Characterization of Wake Vortex

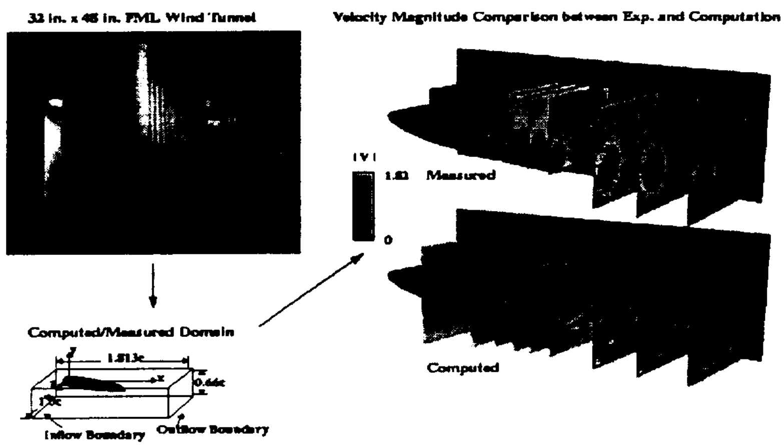




Wing Tip Vortex Validation

Ames Research Center

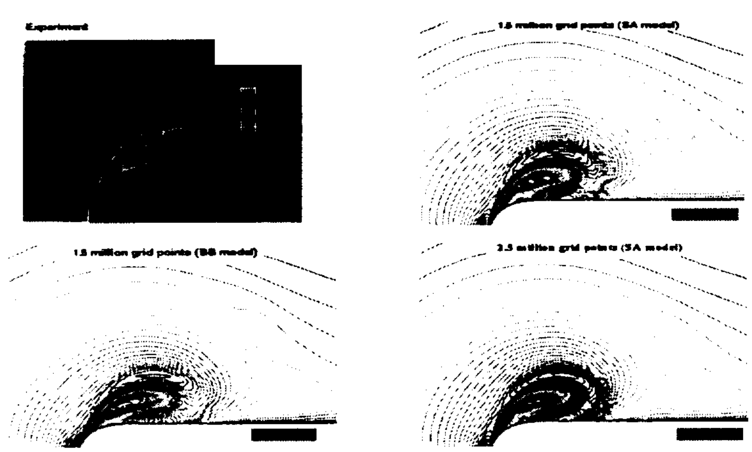
- **Experimental and Computational Model**
  NACA 0012 wing with round tip:Re=4.6×10^{6}, $\alpha=10^{\circ}$



Wing Tip Vortex Validation

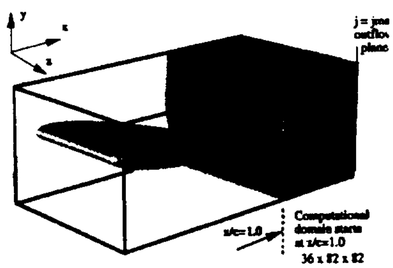
Ames Research Center

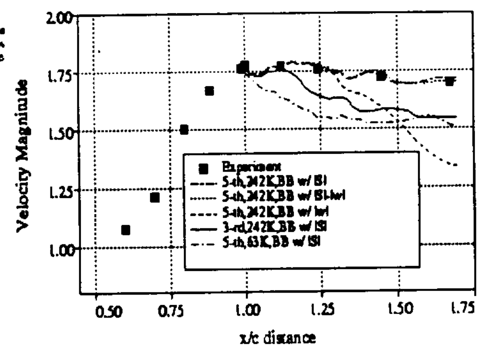
- **Turbulence Model / Grid Density Effect on Roll Up**
  NACA 0012 wing with round tip:Re=4.6×10^{6}, $\alpha=10^{\circ}$



Wake Vortex Validation Using INS3D

Axial progression of wingtip vortex in near-wake
NACA 0012 wing with round tip:Re=4.6×10^6, $\alpha=10°$

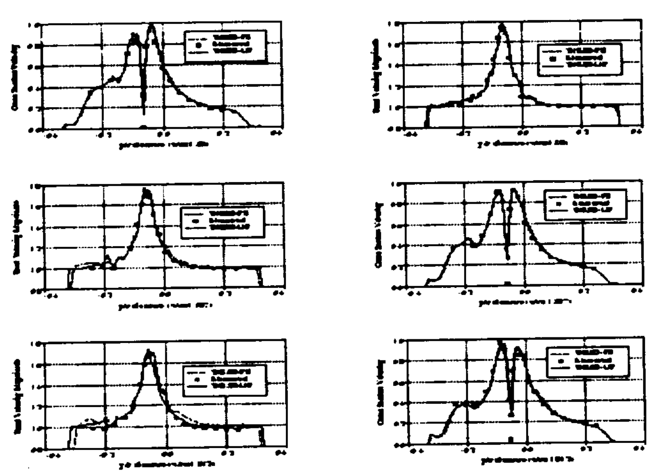




Wake Vortex Validation

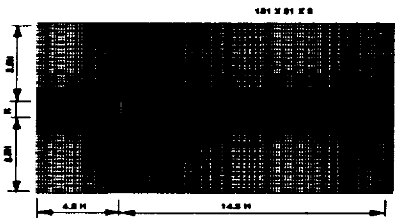
- **Velocity Magnitude at Different Downstream Locations**

  Axial progression of wingtip vortex in near-wake
  NACA 0012 wing with round tip:Re=4.6×10^6, $\alpha=10°$



Impulsively Started Flat Plate at 90°

- GRID
  Thickness of plate = 0.3H

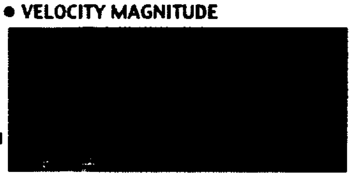


T=1.2

T=2.0

- VELOCITY MAGNITUDE

T=0.4

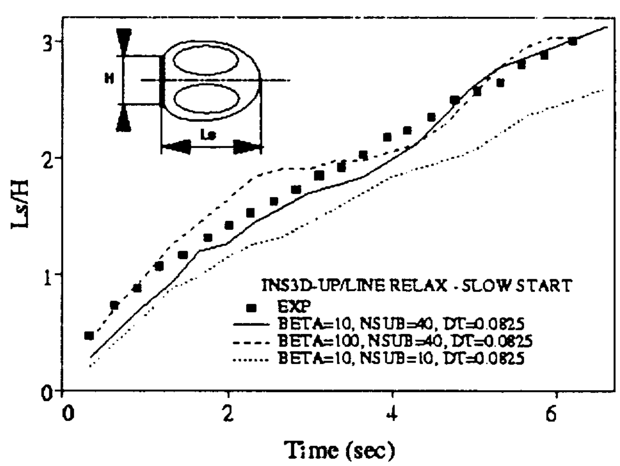


T=4.0

Ref: Issues related to impulsively started problems-
See Gresho and Sani, 1998, CFD Rev 1998
(Hafez/Oshima)

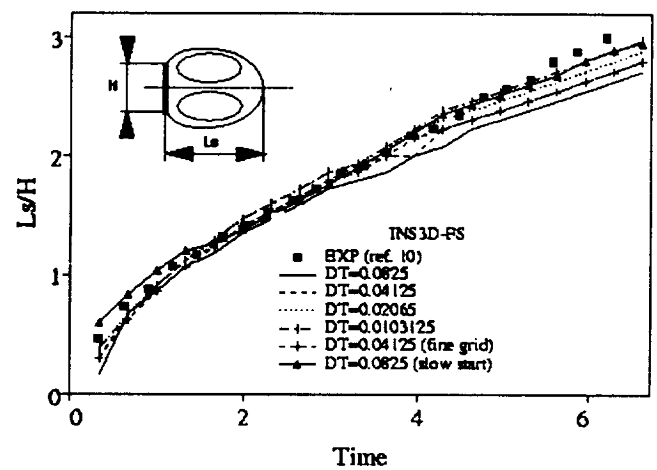
Impulsively Started Flat Plate at 90°

- Time History of Stagnation Point
  Artificial compressibility : Line Relaxation



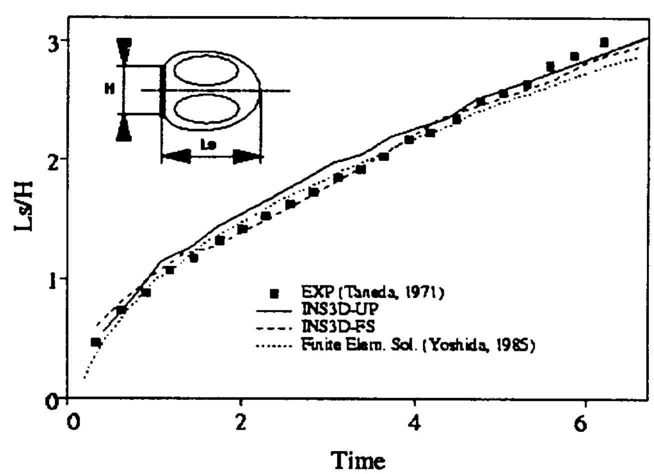
Impulsively Started Flat Plate at 90°

- Time History of Stagnation Point
  Pressure Projection Method (INS3D-FS)



Impulsively Started Flat Plate at 90°

- Time History of Stagnation Point



Vorticity-Velocity Method

Fasel (1976), Dennis et al. (1979), ... , Hafez (1988)

⊙ Momentum equation is replaced by vorticity transport equation

$$\frac{\partial \omega_i}{\partial t} + \frac{\partial \omega_i u_j}{\partial x_j} = \omega_j \frac{\partial u_i}{\partial x_j} + \nu \nabla^2 \omega$$

where the vorticity ω is defined as

$$\omega = \nabla \times \mathbf{u}$$

⊙ Taking curl of the above and using continuity equation,

$$\nabla^2 \mathbf{u} = -\nabla \times \omega$$

⊙ These are solved for velocity and vorticity.

⇒ Computational efficiency of this approach depends on the Poisson solver. Overall performance in general three-dimensional applications remains to be seen.

Engineering Applications

- Applications
  - Viscous incompressible flows are encountered in many applications
  - Computational tools offer greater flexibility for resolving engineering problems, i.e. developing/designing flow devices
  - Issues in engineering applications are illustrated by presenting the following two real-world applications problems
- Turbopump
  - Most of the material was present at
    the First MIT Conference on Computational Fluid and Solid Mechanics, Cambridge MA, June 12-14, 2001, entitled as "High-End Computing for Incompressible Flows"
- Biofluid
  - Most of the material was present at
    the Sixth U.S. National Congress on Computational Mechanics, Dearborn, MI August 1-4, 2001, entitled as "Computational Hemodynamics Involving Mechanical Devices"

Applications Point of View

- Applications to Real-World Problems
  - N-S solution of full configuration was a big goal in the 80s
  - Numerical procedures and computing hardware have been advanced enabling simulation of complex configurations
- Some Examples of Successful Applications
  - Components of liquid rocket engine
  - Hydrodynamics (Submarines, propellers, ...)
  - Ground vehicles (automobile aerodynamics, internal flows...)
  - Biofluid problems (artificial heart, lung, ...)
  - Some Earth Science problems
- Current Challenges
  - For integrated systems analysis, computing requirement is very large
    $\Rightarrow$ Analysis part is still limited to low fidelity approach
  - For high-fidelity analysis, especially involving unsteady flow, long turn-around time is often a bottle neck $\Rightarrow$ Acceleration of solution time is required

Artificial Compressibility Method (INS3D-UP)

- Time accuracy is achieved by subiteration
  - Discretize the time term in momentum equations using second-order three-point backward-difference formula

$$\frac{3q^{n+1} - 4q^{n} + q^{n-1}}{2\Delta t} = -(rhs)^{n+1}$$

  - Introduce a pseudo-time level and artificial compressibility,
  - Iterate the equations in pseudo-time for each time step until incompressibility condition is satisfied.

$$\frac{1}{\Delta\tau}(p^{n+1,m+1} - p^{n+1,m}) = -\beta q^{n+1,m+1}$$

$$\frac{1.5}{\Delta\tau}(q^{n+1,m+1} - q^{n+1,m}) = -(rhs)^{n+1,m+1} - \frac{3q^{n+1,m} - 4q^{n} + q^{n-1}}{2\Delta t}$$

- Code performance
  - Computing time : 50-120 ms/grid point/iteration
  - Memory usage: **Line-relaxation** 45 words/grid point
    GMRES-ILU(0) 220 words/grid point

Pressure Projection Method(INS3D-FS)

- Approach in generalized coordinates
  - Finite volume discretization
  - Accurate treatment of geometric quantities
  - Dependent variables - pressure and volume fluxes
  - Implicit time integration
  - Fractional step procedure
    Solve auxiliary velocity field first,
    then enforce incompressibility condition by solving a Poisson equation for pressure.
- Code performance
  - Computing time : 80 ms/grid point/iteration
  - Memory usage: 70 words/grid point

History of INS3D Development

- Code

1982-1987 Original version
(Kwak, Chang)

1988-1997 INS3D-UP
(Rogers, Kiris, Kwak)
INS3D-LU
(Yoon, Kwak)
INS3D-FS
(Rosenfeld, Kiris, Kwak)

1998-Present Combined version
(Kiris, Kwak)

- Applications

SSME Phase II+
HGM Redesign

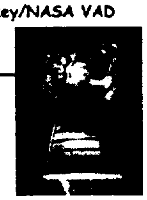
Flight engines
1995

Inducer

Impeller

Human implantation, 1998
DeBakey/NASA VAD

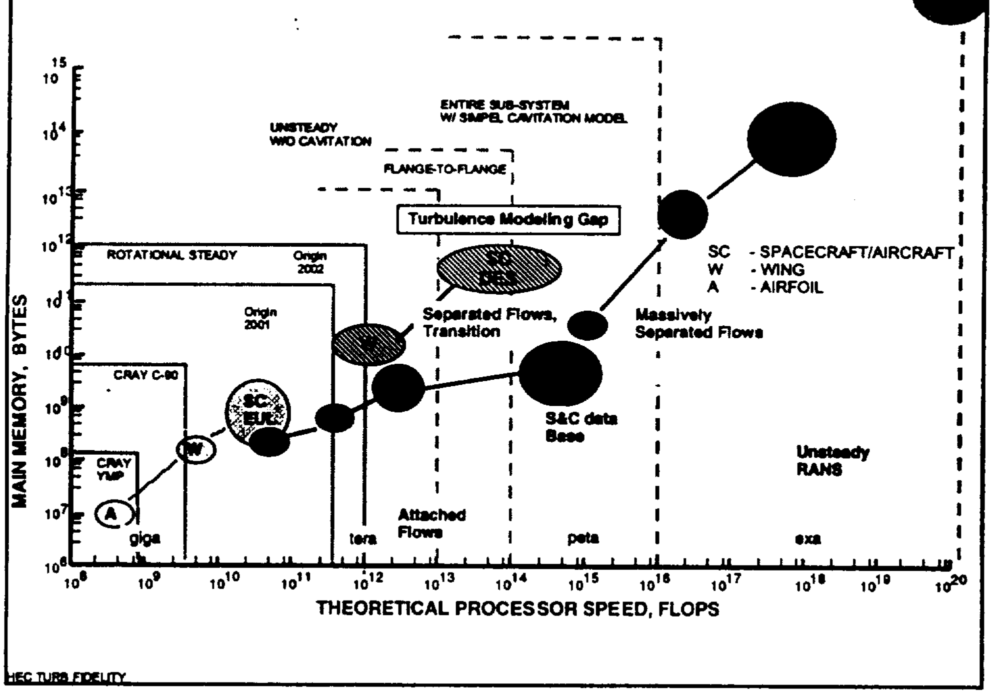
Advanced liquid
sub-systems



Current Challenges

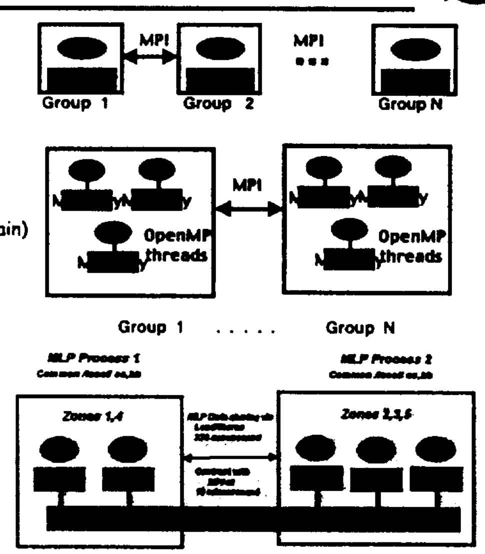
- Challenges where improvements are needed
  - Time-integration scheme, convergence
  - Moving grid system, zonal connectivity
  - Parallel coding and scalability
- As the computing resources changed to parallel and distributed platforms, computer science aspects become important such as
  - Scalability (algorithmic & implementation)
  - Portability, transparent coding etc.
- Computing resources
  - "Grid" computing will provide new computing resources for problem solving environment
  - High-fidelity flow analysis is likely to be performed using "super node" which is largely based on parallel architecture

Supercomputing Memory and CPU Requirements



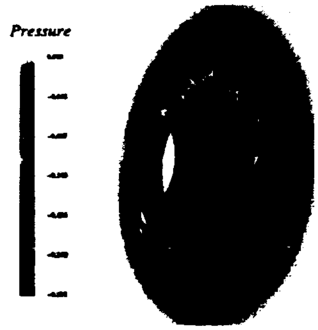
Parallel Implementation of INS3D

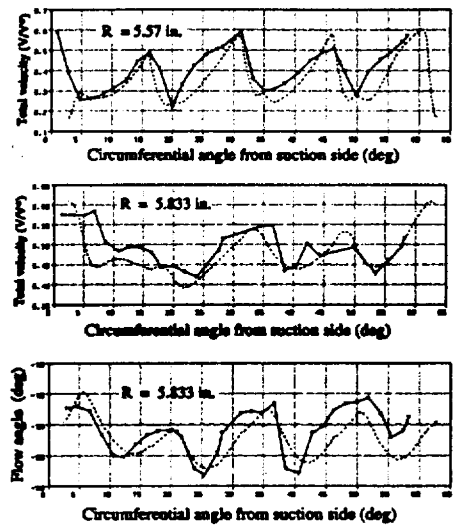
- INS3D-MPI
  (coarse grain)
  T. Faulkner & J. Dacles
- INS3D-MPI / Open MP
  MPI (coarse grain) + OpenMP (fine grain)
  Implemented using CAPO/CAPT tools
  H. Jin & C. Kiris
- INS3D-MLP
  C. Kiris



Parallel Implementation of INS3D

- Previous Work (SSME Impeller)



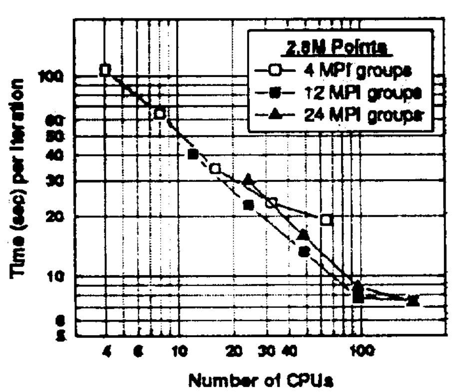


Parallel Implementation of INS3D

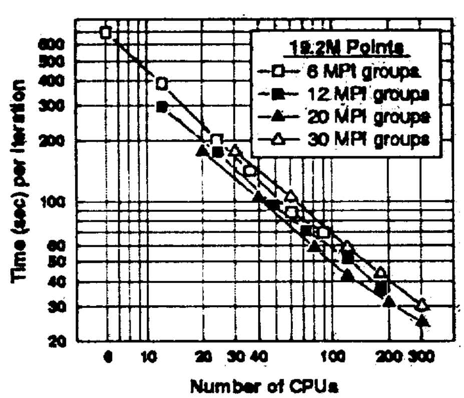
MPI coarse grain + OpenMP fine grain

TEST CASE : SSME Impeller

24 zones / 2.8 Million points



60 zones / 19.2 Million points

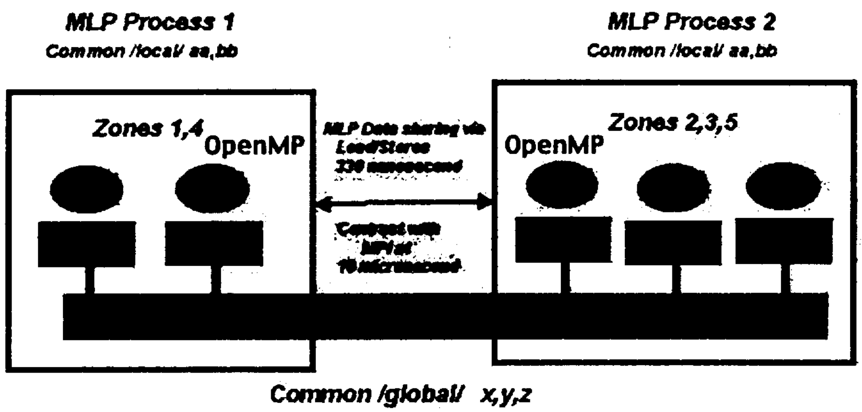


Parallel Implementation of INS3D

Multi-Level Parallelism (MLP)

INS3D-MLP : MLP routines + OpenMP

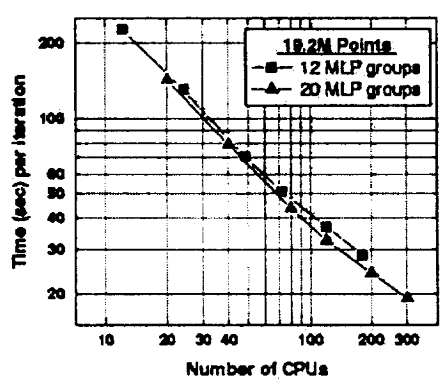
Shared Memory MLP Organization for Origin 2000

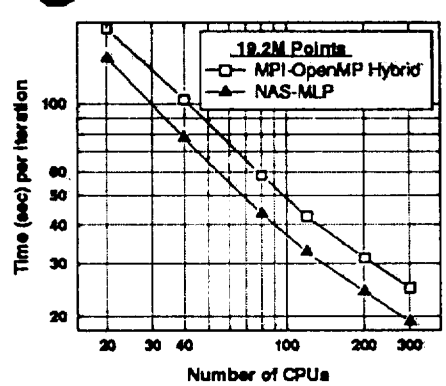


Parallel Implementation of INS3D

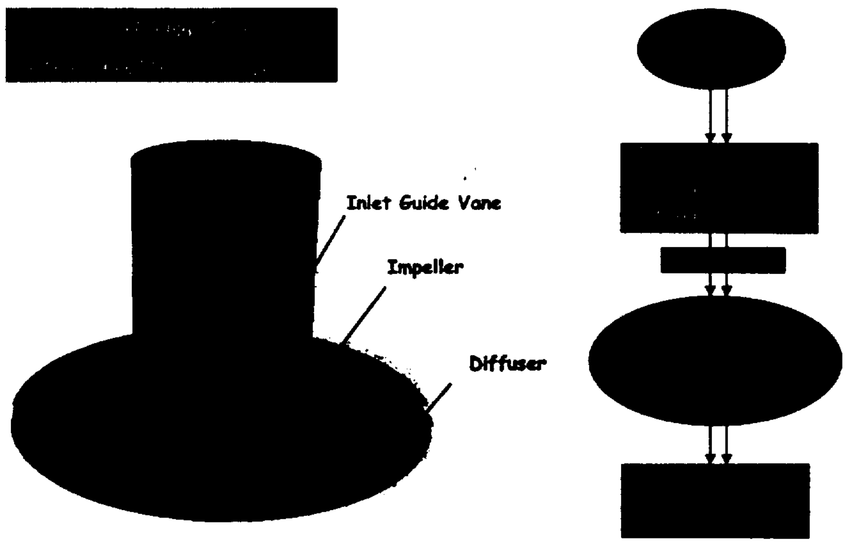
INS3D-MLP (NAS MLP no pin-to-node) / OpenMP

TEST CASE : SSME Impeller
60 zones / 19.2 Million points



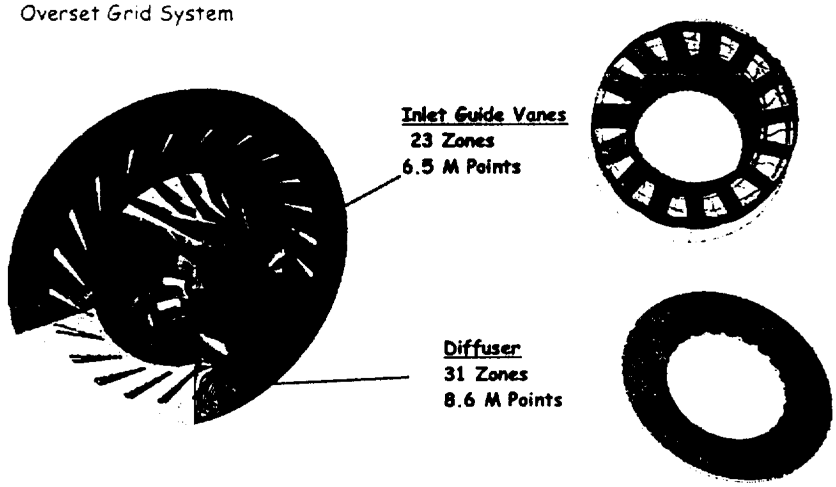


A Generic Rocket Engine Turbopump

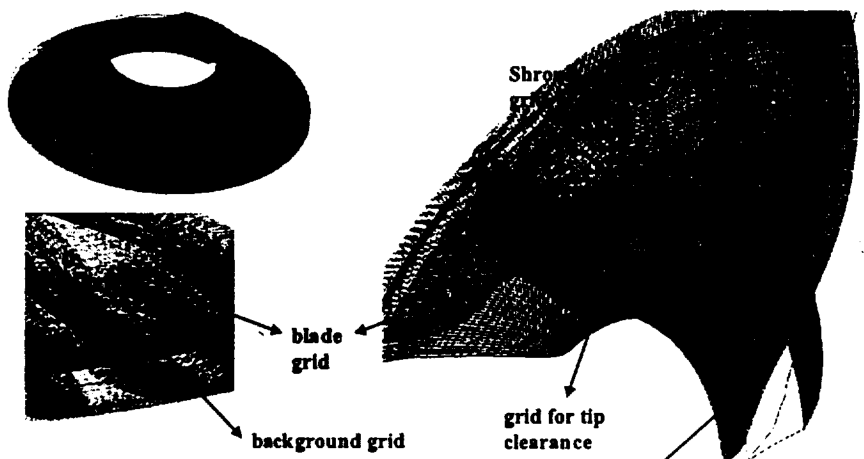


A Generic Rocket Engine Turbopump

Overset Grid System



A Generic Rocket Engine Turbopump

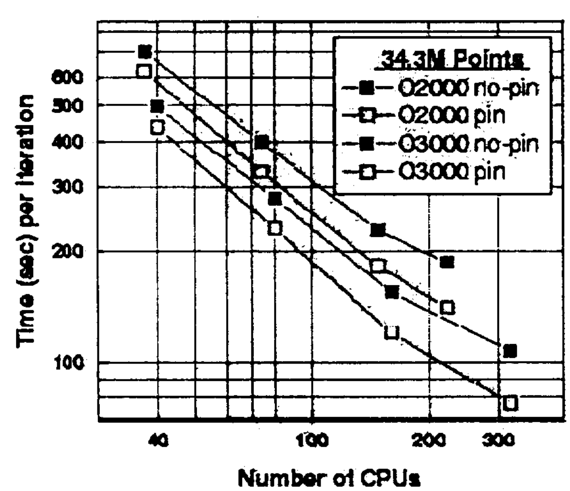


Impeller Grid :
60 Zones / 19.2 Million Grid Points
Smallest zone : 75K /Largest zone : 996K
Less than 192 orphan points.

Parallel Implementation of INS3D

INS3D-MLP / 40 Groups

Turbo pump
114 Zones / 34.3 M grid points

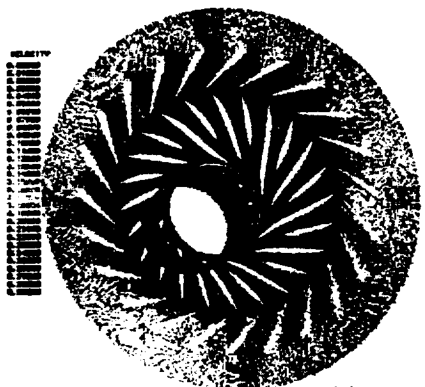


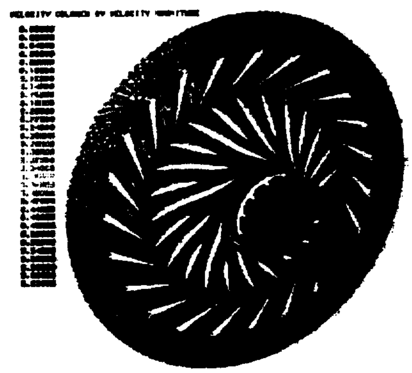
Per processor Mflop is between 60-70.
Code optimization for cache based platforms is currently underway.
Target Mflops is to reach 120 per processor.
Increasing number of OpenMP threads is also the main objective for this effort.

Overset Grid System for A Turbopump

Initial Start: first time step

Time Step 5





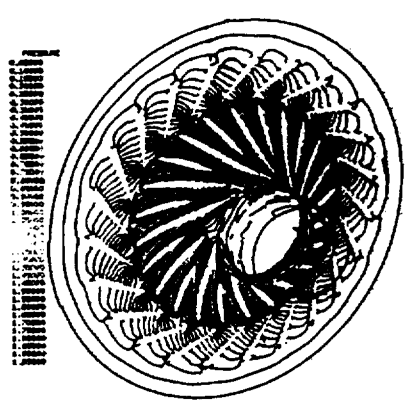
Velocity colored by magnitude

Impeller started at 10% of design speed

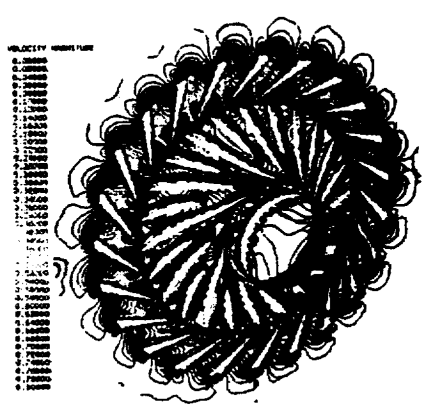
Impeller rotated 2.25 degrees

Initial start

Time Step 7: Impeller rotated 3-degrees at 30% of design speed



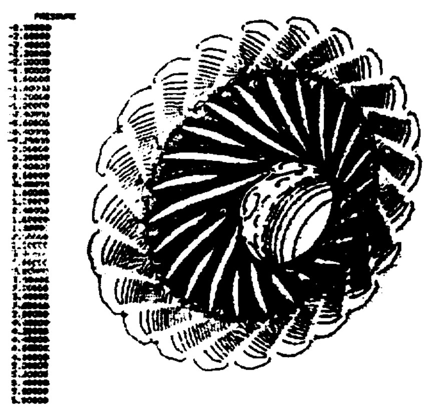
PRESSURE



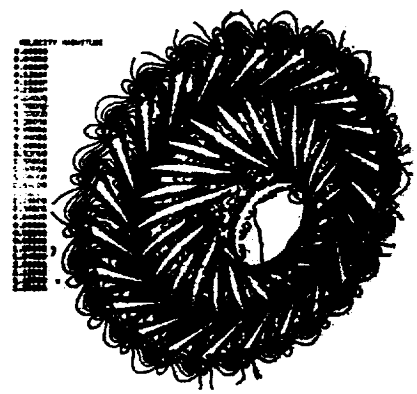
VELOCITY MAGNITUDE

Initial start

Time Step 18: Impeller rotated 8-degrees at 100% of design speed

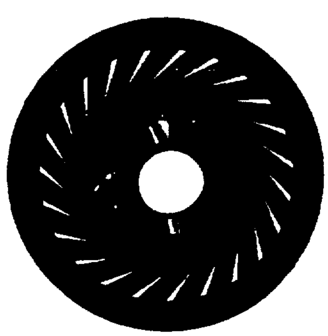


PRESSURE



VELOCITY MAGNITUDE

Summary of A Generic Turbopump Simulation



Problem size:
- 34.3 Million Points
- 800 physical time steps in one rotation

CPU requirement:
- One physical time-step requires less than 20 minutes wall time with 80 CPU's on Origin 2000.
- One complete rotation requires one-week wall time with 80 CPUs.

I/O:
- Currently I/O is through one processor. Timing will be improved with parallel I/O since time-accurate computations are I/O intensive

Parallel Efficiency:
MLP/OpenMP version requires 19-25% less computer time than MPI/OpenMP version.
Pin-to-node for MLP version reduces computer time by 40%

Discussion on Numerical Procedures

- Finite Difference
  - Based on Taylor series expansion ⇒ Requires smooth grid
  - Need special care for grid singularity
  - Generally easier to use fine grids near wall at high Reynolds number
- Finite Volume
  - Formulation is more physical (conservation of properties)
  - Viscous flux calculation is not as straightforward
  - Difficult to implement higher order schemes
- In actual implementation, however,
  - These differences become unclear
    i.e. - FV in curvilinear coordinates requires lots of averaging depending on definition of variables such as staggered vs cell vertex arrangement
    - Both FD and FV implementations are very similar near grid singularities
  - Major differences come from time integration scheme which also affects the computational efficiencies, especially, for unsteady flow computations

Discussion on Applications

- Rapid turn around can be accomplished through the use of
  - Algorithm : convergence acceleration such as multi-grid, and GMRES
  - Parallel implementation
- Total process time can be reduced by
  - Automatic solution process including CAD to grid procedure
- Need further development of methodology as well as physics modeling for
  - Deep understanding of flow physics such as unsteady flow characterization for better aeroacoustics modeling,and flow induced vibration
    ⇒ Is LES method mature enough for this?
  - Need to matrix IT tools to flow simulation for smart flow control and optimization
- Efficient extraction of information is still a challenge

On top of all these we still need trained CFDers to solve many unsolved real world problems, for development of flow devices and for better understanding of flow physics

Applications to Biofluid Devices

- Motivation

- Mechanical Heart Assist Devices
  - Computational Issues and Requirements
  - Pulsitile Device
  - Axial Flow Pump

- Computational Approach for VAD Development
  - CFD Technology Developed for Space Shuttle
  - Design Improvements Using CFD

- Summary and Discussion

Mechanical Assist Devices

- Motivation
  - Over 5 million Americans and 15 million people worldwide suffer from Congestive Heart Failure (CHF)
  - CHF patients are still treated with drug therapy, however, at late stage heart transplantation is traditionally the only treatment hope
  - Mechanical heart assist devices are being used as a temporary support to sick ventricle and valves as a

    "BRIDGE-TO-TRANSPLANT" or "BRIDGE-TO-RECOVERY"

  - Need for assist devices is very high

Permanent VAD need	: 25,000-60,000 / YR
Current valve replacement	: 120,000/ YR
Donor hearts available	: 2,000-2,500 / YR

Mechanical Heart-Assist Device

- Heart Valves
- Ventricular Assist Device (VAD)

Pulsatile Pump
- Piston Driven : Low speed, Bulky
- Pneumatically Driven : Need external support equipment

Rotary Pump
- Axial Flow Pump : High speed, Small

⇒ **DeBakey VAD is based on this concept**

- Total Artificial Heart

Ventricular Assist Device

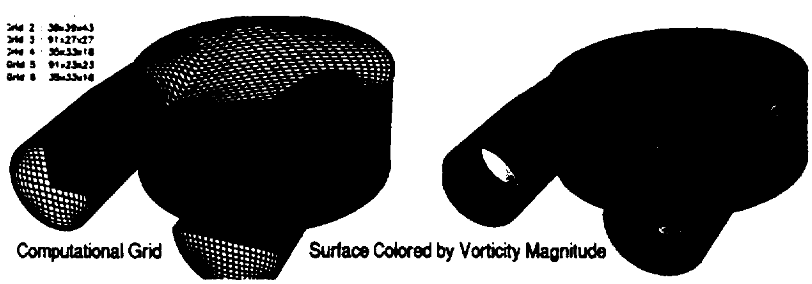
- Requirements
  - Simplicity and Reliability
  - Small size for ease of implantation
  - Supply 5 liter/min of blood against 100 mmHg pressure
  - High pumping efficiency to minimize power requirements
  - Minimum Hemolysis and Thrombus Formation

Computational Issues

- Geometry / grid definition
  Moving boundary, flexible wall
- Solver
  Time accurate solver
- Physical modeling
  Newtonian vs non-Newtonian
  Turbulence
- Experimental & clinical data

Example of Pulsatile Pump

- Penn-State Artificial Heart
  Chimera Grid for moving components

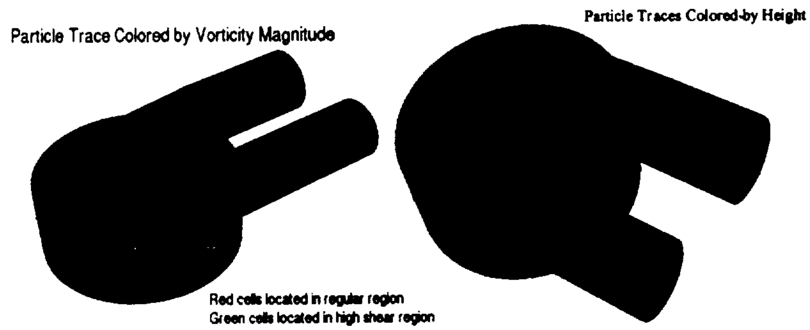


Computational Grid   Surface Colored by Vorticity Magnitude

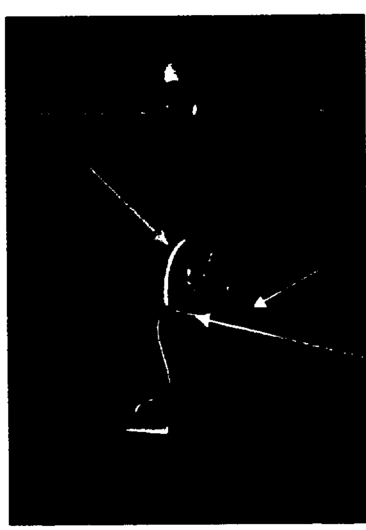
This and other results were first reported by Kiris et. al in 1991:
"Computation of Incompressible Viscous Flows through Artificial Heart Devices with Moving boundaries," Proc. American Mathematical Society Summer Research on Biofluid Dynamics Conference, July 6-12, 1991, Seattle, WA

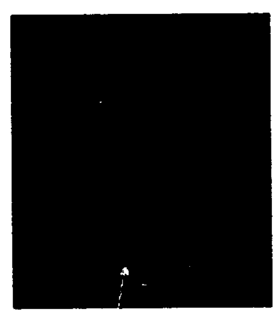
Example of Pulsatile Pump

- Penn-State Artificial Heart
  Analysis of time dependant data was an issue

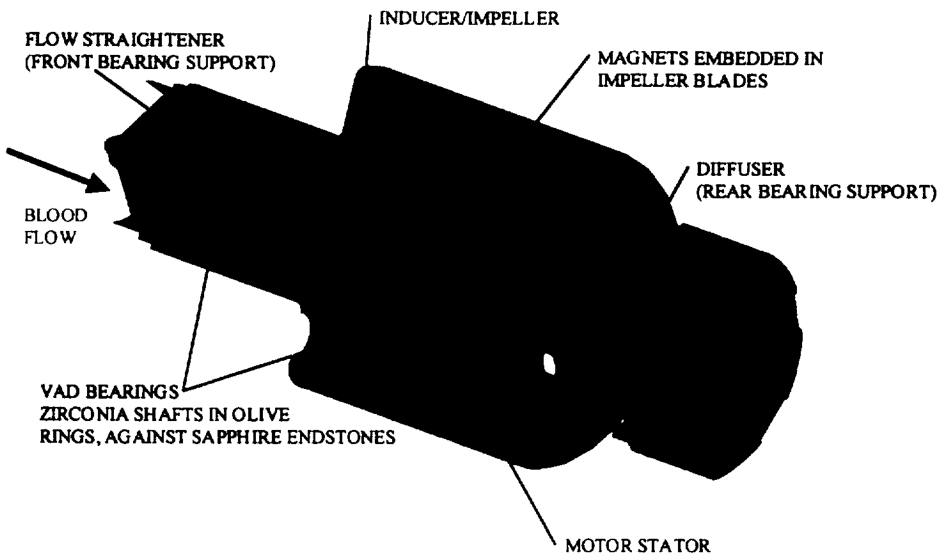


Schematic of DeBakey VAD™





DeBakey VAD™ Pump Schematic



Issues in Axial flow VAD

- Problems Related to Fluid Dynamics
  - Small size requires high rotational speed
    Highly efficient pump design required

  - High shear regions in the pump may cause excessive blood cell damage
    Minimize high shear regions

  - Local regions of recirculation may cause blood clotting
    Good wall washing necessary

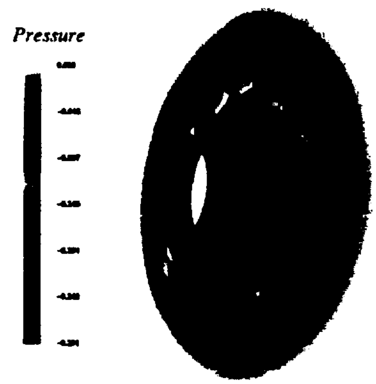
⇒*Small size and delicate operating conditions make it difficult to quantify the flow characteristics experimentally*

Validation-SSME Turbopump Flow Analysis

- SSME HPFTP 11' Impeller

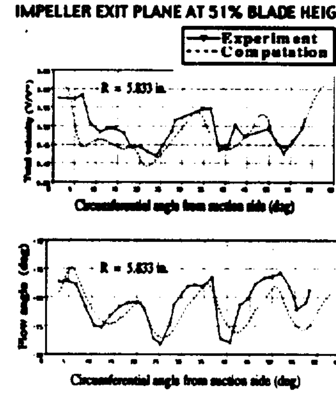
Shrouded impeller: 6 full blades, 6 long partials, 12 short partials 6322 rpm, Re=1.81x10⁵ per inch

HUB SURFACE COLORED BY STATIC PRESSURE



COMPARISON WITH EXPERIMENTAL DATA

IMPELLER EXIT PLANE AT 51% BLADE HEIGHT



DeBakey VAD Development Timeline

- Baseline Design

1984 - NASA Johnson Space Center's David Saucier begins initial design work on axial pump VAD with Dr. DeBakey

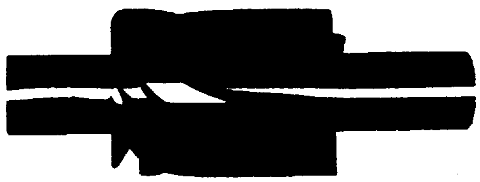
1988 - NASA/JSC and Baylor College of Medicine signs Memorandum of Understanding to develop the DeBakey VAD

1992 - NASA/JSC begins funding the project

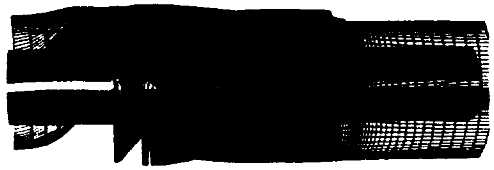
NASA/DeBakey VAD (Baseline Design)

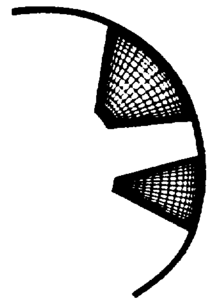
NASA / DeBakey Axial Flow VAD Impeller

Zone 1 :101 x 39 x 33
Zone 2 :101 x 39 x 33
Zone 3 : 59 x 21 x 7
Zone 4 : 47 x 21 x 7
Zone 5 : 59 x 21 x 7



Geometry





Computational Grid

Rotational Speed : 12,600RPM
Flow Rate : 5 lit/min

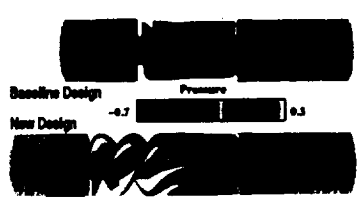
DeBakey VAD Development Timeline

- CFD Assisted Design

1993 - **NASA/ARC is asked to develop CFD procedure to improve design and performance. D. Kwak and C. Kiris visit JSC to study the device**
The technology developed for rocket engine such as the Space Shuttle main engine was to be extended to blood flow simulation

1994 - **Kiris and Kwak begin work on design analysis using NAS supercomputers**

⇒ NEW DESIGN WAS PROPOSED TO INCLUDE AN INDUCER BETWEEN THE FLOW STRAIGHTNER AND THE IMPELLER



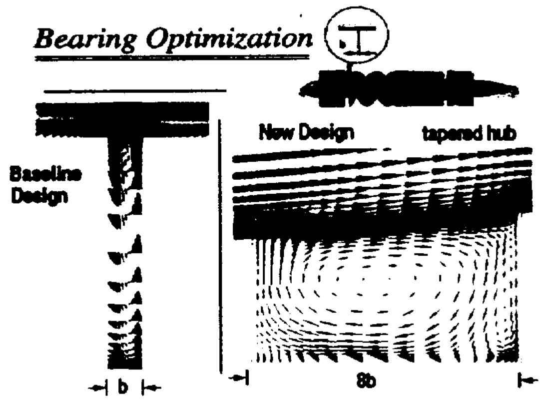
Particle Traces Colored by Velocity Magnitude

DeBakey VAD Development Timeline

- CFD Assisted Design

1994 -Kiris and Kwak continued design changes

⇒ IMPROVE BEARING, HUB AND HUB EXTENSION DESIGN TO REDUCE BLOOD CLOTTING



DeBakey VAD Development Timeline

- Animal Tests

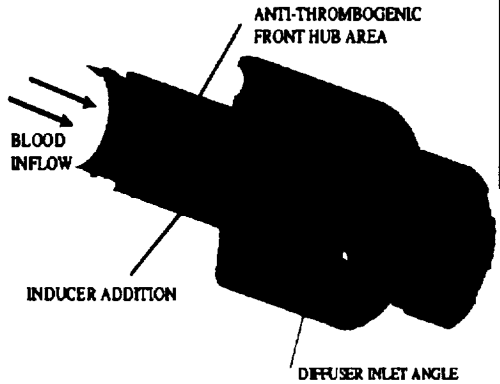
1995 - Animal implantation: passed two-week requirements

1996 - Fulll design rights are granted to MicroMed, Inc. to produce the pump
Began using bio-compatible titanium replacing polycarbonate

1997 - Configuration design finalized

CFD Contributions to VAD Design

CFD Contributions To Design



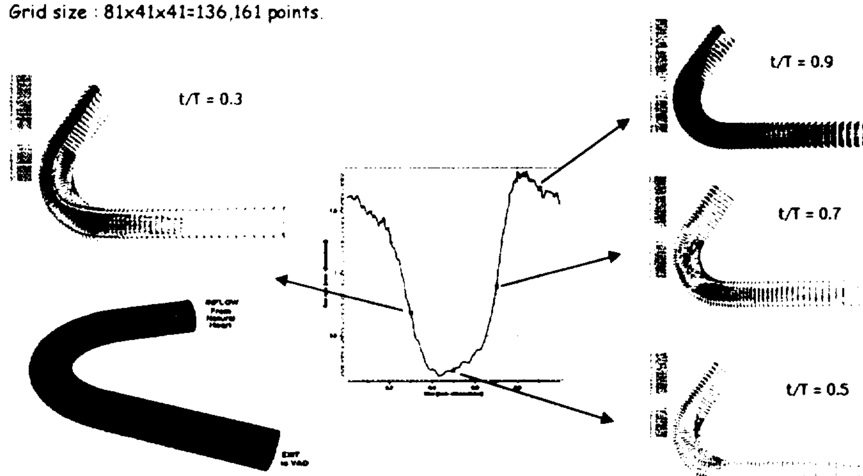
	0.02	0.002
	Yes	no
	2 days	30+ days
		~ 1 year*

* As of July 2001

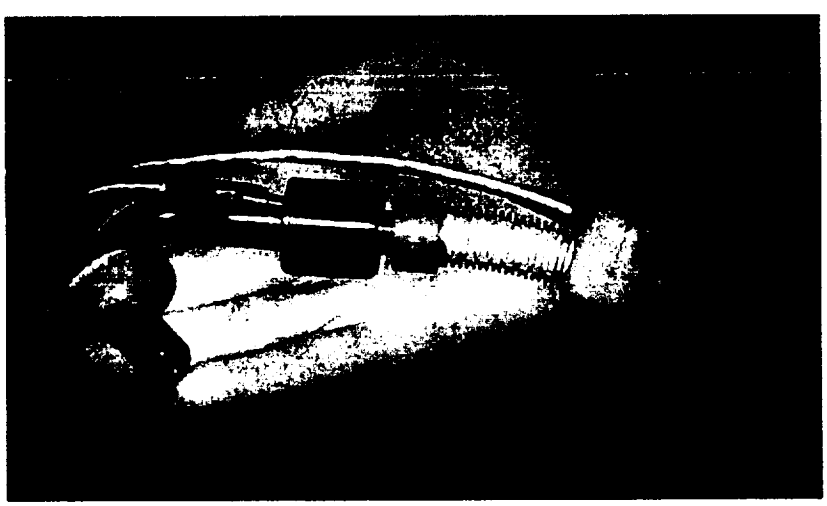
- *Inducer addition*
- *Bearing cavity design*
- *Change diffuser inlet angle*

Inlet Cannula Study

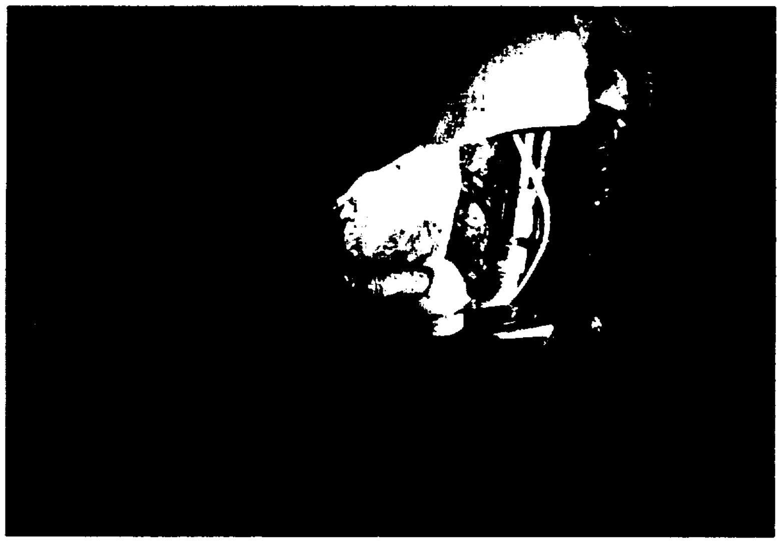
Time-dependent inflow flow rate is used in the elbow parametric study
Grid size : 81x41x41=136,161 points.



DeBakey VAD



DeBakey VAD - In situ



DeBakey VAD Development Timeline

- Human Implantation in Europe

1998 - On November 13, 1998, the first six DeBakey VADs are implanted in European patients by Roland Hetzer and DeBakey at the German Heart Institute of Berlin. One of the patients, fifty six year old Josef Pristov, is able to return home and spend Christmas with his wife after a month's stay for recovery and monitoring at the clinic

VIE 201 & 202 at lunch

DeBakey VAD Development Timeline

- Human Implantation in USA

1999 - US Patent is granted for the device on September 9, 1999

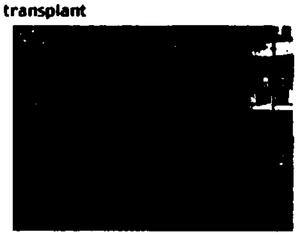
2000 - Over 30 patients have received the device
The longest successful trial period to date in human is approaching 1 year (as of July 2001)

Ames Research Center

NASA/DeBakey VAD
Accomplishments to date (7/1/01)

A patient in Munich fully mobile and discharged awaiting transplant

A patient in Zurich, owner/chef of a restaurant worked daily until his transplant



- 90+ patients implanted
- Number of patients currently ongoing with device (longest patient supported with the device is now approaching one year)
- US trial
  Approved for 20 patients in a multi-center trial
- European trial
  Received "CE mark" (the EU equivalent to FDA approval)
- Results to date
  Favorable compared to existing VADs
  Small incidence of thrombus is being investigated
  ⇒ Further computational support is essential

The first patient in Houston with Drs. DeBakey and Noon on her discharge day after transplant

Ames Research Center

Summary and Discussion

- Computational approach provides
  - a possibility of quantifying the flow characteristics: especially valuable for analyzing compact design with highly sensitive operating conditions
  - a tool for conceptual design and for design optimization
- CFD + rocket engine technology has been applied
  - to modify the design of NASA/DeBakey VAD which enabled human implantaion
- Computing requirement is still large
  - Unsteady analysis of the entire system from natural heart to aorta involves 625 revolutions of the impeller
  - During on heart beat, impeller has 125 revolutions
  - With 1024 processors of Origin, one simulation (with several heart beat) from heart to aorta can be completed in one month
- Further study is needed
  - to assess long term impact of mechanical VAD on human body, which requires modeling <u>flexible wall and non-Newtonian effect</u> among other things
- There exist some gaps between
  - CFD (assuming IT is a part of CFD applications) and biomedical expertise

Current Challenges of CFD

Dochan Kwak
Applications Branch Chief
NASA Advanced Supercomputing (NAS) Division
NASA Ames Research Center
Moffett Field, California

This presentation is my personal opinion
and does not reflect organizational viewpoint

Ames Research Center

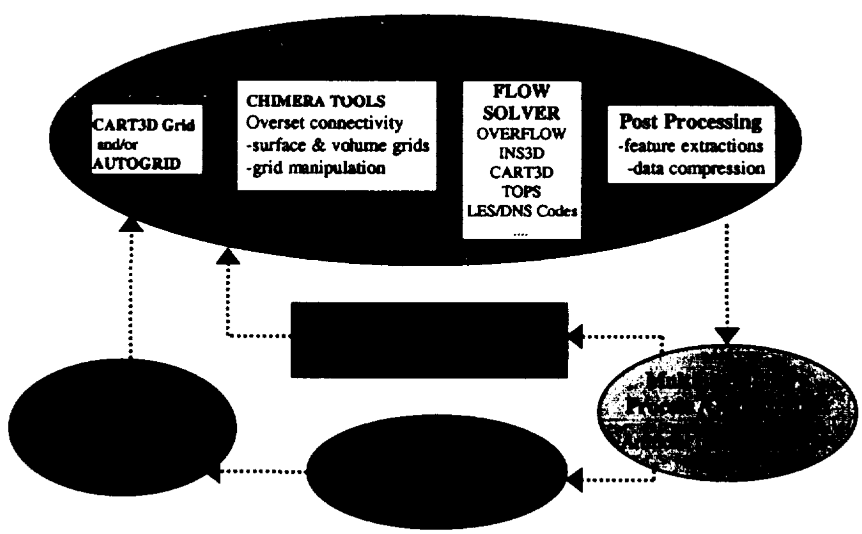
Topics for Discussion

- Progress to date
- Status of CFD Research
- Current need in flow simulation
- Some Challenges

Progress to Date

- CFD has pioneered the field of flow simulation for
  - Obtaining engineering solutions involving complex configurations
  - Understanding physics (critical to mission success)
- CFD has progressed as computing power has increased
  - Numerical methods have been advanced as CPU and memory increase
  - N-S solution of full configuration was a big goal in the 80s
  - Complex configurations are routinely computed now
  - DNS/LES are used to study turbulence, (but not to resolve mysteries of turbulence)
- As the computing resources changed to parallel and distributed platforms, computer science aspects become important such as
  - Scalability (algorithmic & implementation)
  - Portability, transparent coding etc.
  - Coding paradigm is being changed (i.e. object-oriented program...)

Simulation and Control Roadmap



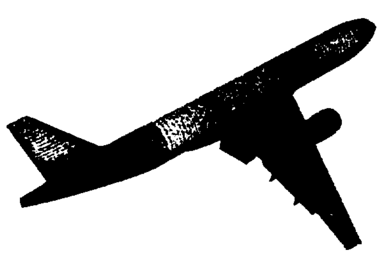
Examples of Current Capability

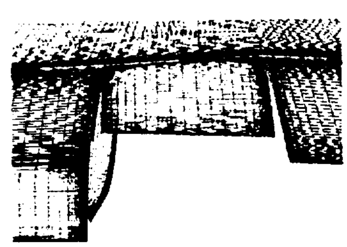
- Algorithmic advances include
  - Discrete models :
    Various artificial dissipation models
    Unified formulations, e.g. preconditioning
    Unstructured methodology
    Various gridding strategies
  - Solution methods:
    Explicit/Implicit
    Preconditioning, dual-time
    Multi-grid
- Successful application of CFD to engineering problems
  - High-lift configurations
  - Multiple bodies in relative motion
  - Components of propulsion system (both aero & space)
  - Maneuvering vehicle
  - .....
  - List goes on

Examples of Current Capability: OVERSET CFD Tools

- EXAMPLE LANDING CONFIGURATION
  - 22.4M mesh points
  - 79 zones
  - 201 C90 hours for convergence
    (Lift within 2% of experiment)

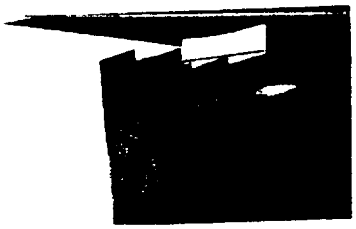
⇒ Small geometric variations have a major impact, particularly near maximum lift
⇒ Grid density study was performed
⇒ Accuracy of physical modeling needs further assessment





Stuart Rogers, NASA Ames - AST/IWD High Lift

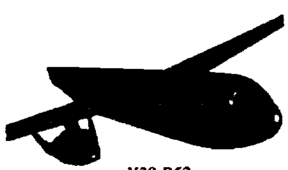
Overset Technology for Complex Configuration



- Overset (Chimera) Grid Approach
  NASA Ames Developed CFD Tools
  - OVERFLOW Navier-Stokes Flow Solver
  - Chimera Grid Tools: Pre- and Post Processing
  - Enabling flow simulation technology for complex configuration and unsteady flow involving bodies in relative motion

- OVERFLOW+CGT: 1996 NASA Software of the Year Honorable Mention
- PLOT3D Visualization Software
  1993 NASA Space Act Award
- FAST Visualization Software
  1995 NASA Software of the Year Award

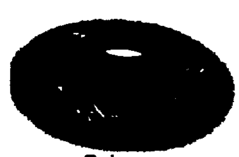
Overset Technology for Complex Configuration



X38 B52
Drop

High-Lift
Aircraft
in Wind Tunnel

- Wide Range of Applications
  - Spacecraft ascent and descent (still need better chemistry)
  - Propulsion (limited scope)
  - Aircraft
  - Hydrodynamics (limited unsteady capability)
- Current Development
  - Working toward fully automated grid generation
  - Steady and Unsteady capabilities (in all speeds)
  - Bodies in Motion, 6 DOF
  - Non-equilibrium chemistry

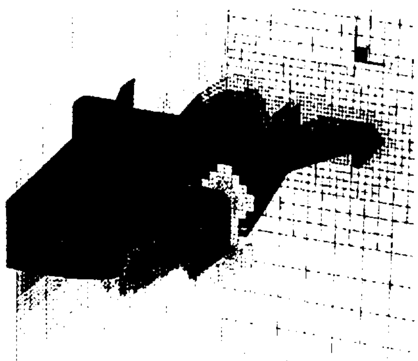


Turbopump

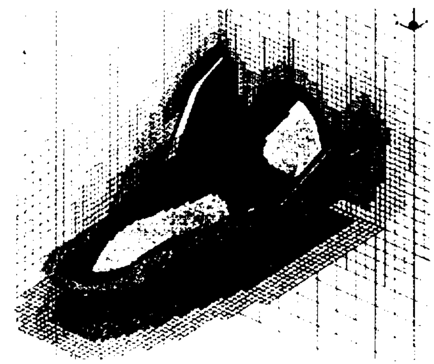
Cart3D - Cartesian-Based Euler Design Tool

- Example Meshes

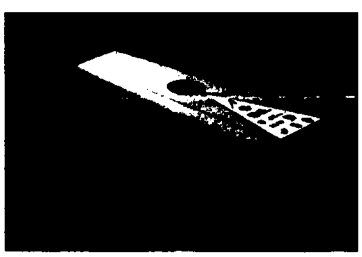
Generic SSTO : 1.2M Cells



X-38 : 1.6M Cells



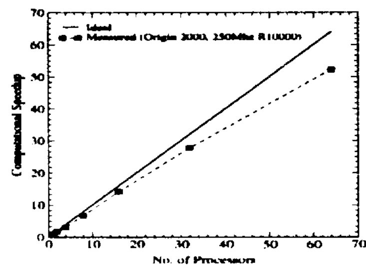
Cart3D : Conceptual Design Evaluation Tool



- Enables rapid conceptual design
  - 180 Cases delivered in 18 days using a desktop machine
  - Test matrix
    Mach 0.3 ~ Mach 5
    Alpha -5deg ~ +15deg
    beta -1deg ~ +1deg
    3 Asymmetric bodyflap deflections

Simulations performed by S. Pandya, Ames Research Center

Scalability of Parallel *Cart3D*

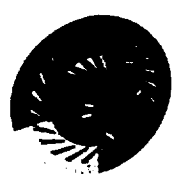


- Parallel Version of Cart3D
  - **Parallelization through domain decomposition and explicit message passing.**
  - **Domain decomposition technique based upon space-filling curves permits domain decomposition to be performed in parallel and on-the-fly (at runtime)**
  - **Parallel speed-ups of ~53 on 64 processors.**
  - **Combined with robust multigrid to offer exceptionally fast convergence across the range of Mach numbers.**
- Validation using Citation Twin-engine Business Jet
  - **1.42M cells, Mach 0.84 alpha 1.81deg**

Unsteady Simulation of A Generic Turbopump

Impeller

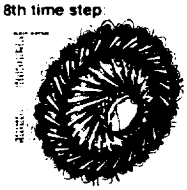
Overset grid set up



1st time step

Velocity vectors:
Impeller started at 10% actual rotational speed.

18th time step



Velocity vectors:
Impeller rotated 8-degrees at 100% of design speed

- Status
- 34.3 Million Points
- 800 physical time steps / rotation

⇒One physical time-step requires less then 20 minutes wall time with 80 CPU's on Origin 2000. One complete rotation requires one-week wall time with 80 CPUs.
⇒ Currently I/O is through one processor. Timing will be improved with parallel I/O since time-accurate computations are I/O intensive.

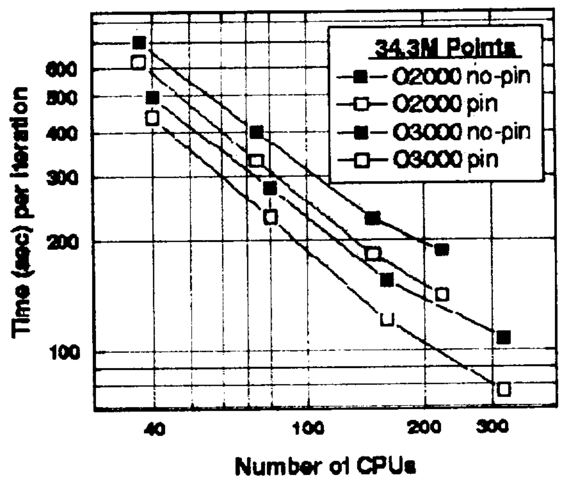
- Current Goal
- Further automation of moving grid
- Parallel I/O

⇒ One rotation per day

Parallel Implementation of Flow Solver

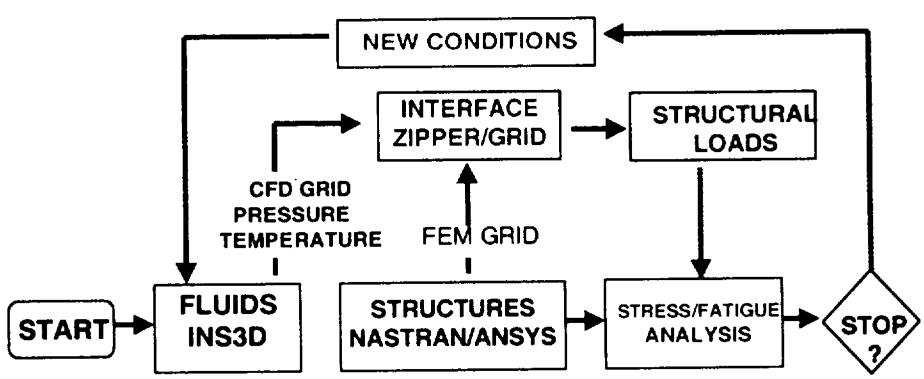
INS3D-MLP / 40 Groups

A Generic Turbo pump
114 Zones / 34.3 M grid points



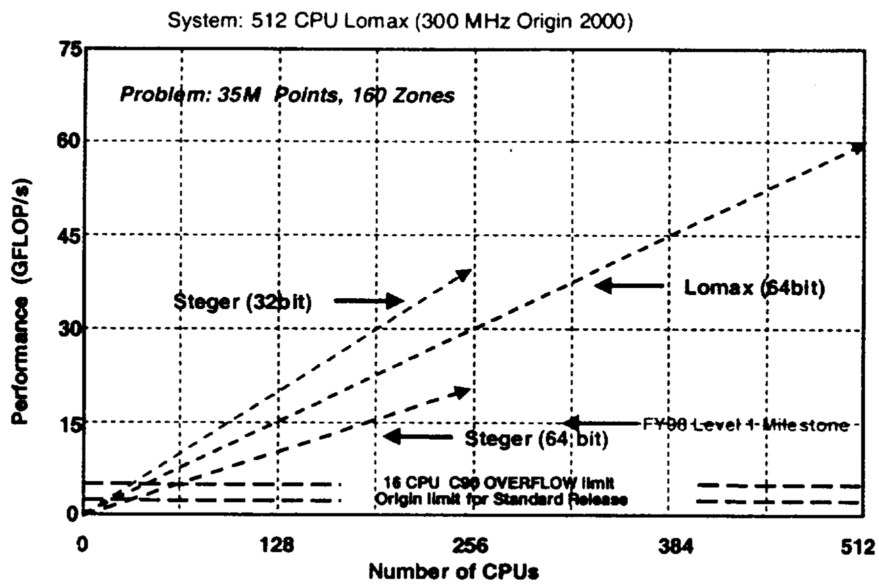
Per processor Mflop is between 60-70.
Code optimization for cache based platforms is currently underway.
Target Mflops is to reach 120 per processor.
Increasing number of OpenMP threads is also the main objective for this effort.

Static/Dynamic Stress Analysis for Flow-Induced Vibration



USE PSE/IPG TOOLS (GROWLER/SCIRUN/PUNCH)

Examples of Current Capability: OVERFLOW-MLP Performance



Examples of Current Capability: OVERFLOW-MLP Performance

- Origin 2000 (64 bit) performance is dramatically better than full C90
  - OVERFLOW 16 CPU C90 = 4.6 GFLOP/s
  - OVERFLOW 256 CPU O2K (250MHz) = 20.1 GFLOP/s
  - OVERFLOW 512 CPU O2K (250MHz) = 37.0 GFLOP/s (cluster)
  - OVERFLOW 512 CPU O2K (300MHz) = 60.0 GFLOP/s
- Performance/Cost Advantage of Steger/Lomax over C90
  - OVERFLOW = 256 CPUs are 4.4x faster @ 4.5x Cheaper = 23x
  - OVERFLOW = 512 CPUs are 13.0x faster @ 2.6x Cheaper = 33x
- Performance gains for small changes in code
  - ~1000 lines of changes (<1% of total code)

Status of CFD Research

"Can do it all" message was propagated in the past, but
CFD did not replace Wind Tunnel ⇒ CFD was oversold!

Of course, further research will create advances with across the board benefits;

- Algorithm
  Convergence acceleration, Robustness, Error estimation
  Grid related issues, adaptive grids ....
- Physical modeling issues
  Turbulence, Combustion, Multiphase, Cavitation, Spray, Plasma etc.
- Solution Procedures
  Automation: CAD-Grid-Solution-Feature extraction
- Applications
  Rapid turn around for complex configurations
  Design and product development - we still need trained CFDers
  ⇒ Outsourcing makes sense

However, sponsors are likely to view these as "incremental advances."

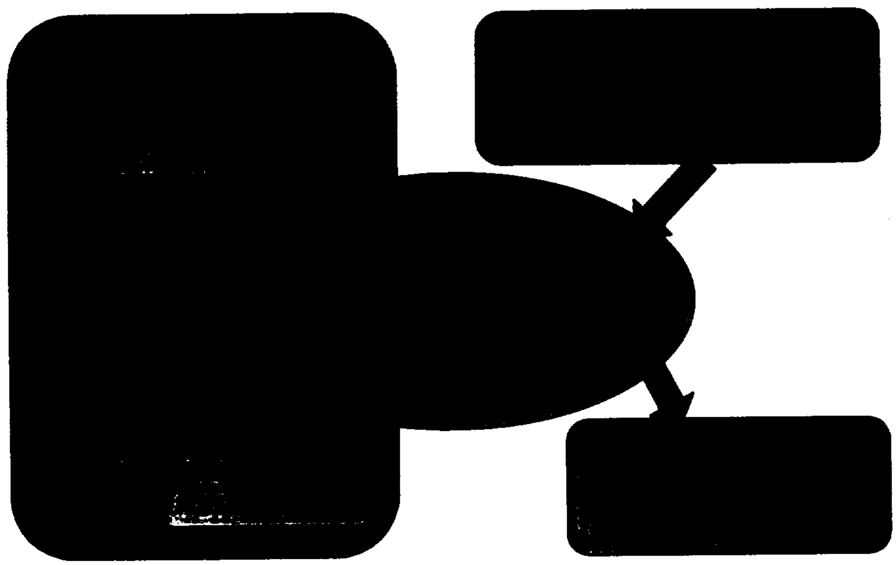
Current Needs in Flow simulation

- Work environment is different now
  - Tremendous information is available
  - Single-handed code development is rapidly becoming outdated (CFD discipline as defined in the past is disappearing)
  - Problem solving environment is more collaborative

  ⇒ Requires software engineering to mitigate risks:
  Legacy software handling tools
  Visualization
  Data base handling tools
  Object-oriented coding

Ames Research Center

Example of Problem Solving Environment



Ames Research Center

Example of Data Base Management Tool: Data Compression Using Multi-resolution

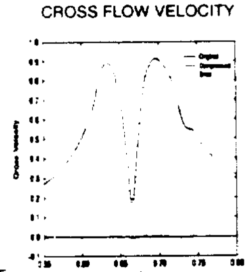
- Wing Tip Vortex Validation
  NACA0012, A R=0.75, Re=4.6x106, α=10°
  INS3D Code, 2.5M Grid (115x189x115)

IMAGES BEFORE AND RECONSTRUCTED FROM COMPRESSED DATA ARE INDISTINGUISHABLE

Compression Ratio :
40 (Pressure), 45 (Pressure & Velocities)
Error: 7.93x10^{-2} (Max Residual), 2x10^{-6} (L_2)

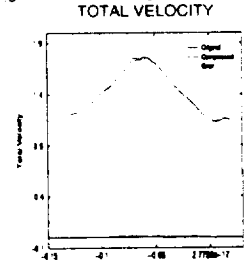
Computation by Jennifer Dacles-Mariani
Data Compression by Dohyung Lee

CROSS FLOW VELOCITY



X/C=1.5

TOTAL VELOCITY



Technology Need

- Integrated solution for assessing the total system performance, life cycle and safety can very well be the next challenge
  e.g. Need a more complete picture of entire design space not just one design

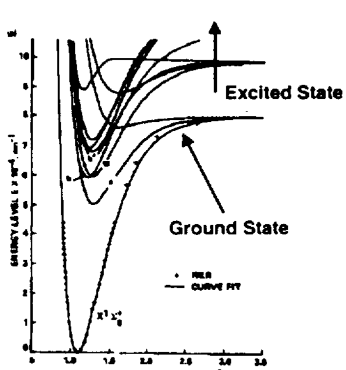
  Some challenges specific to CFD are:

  - **Physics-based simulation for more predictive capability**
  - **Integrated analysis**
    e.g. multi-discipline, performance for entire flight envelope
  - **IT tools can be used to integrate CFD, experiments and flight tests**
    e.g. virtual flight
    ⇒ Requires : Many simulations which will be put into data base, and data base management tools, query tools to extract desired info

- Validation is an issue

Example: Real Gas Effect Model

ELECTRONIC STATE FOR N2



- *Current Model*

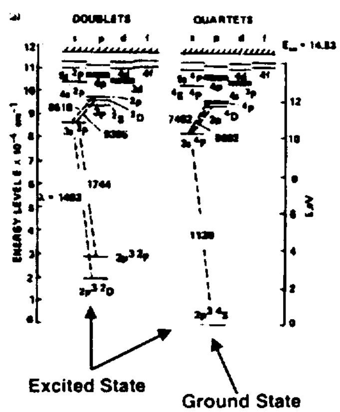
  Euler :
  - **Do not require knowledge of internal molecular structures and intermolecular potentials**

  Navier-Stokes :
  - **Molecules are structureless**
  - **Transport properties are based on a single intermolecular potential**
  - **Collisions are assumed to be elastic**

  Non-equilibrium flow equations:
  - **EOS for each species is based on equil distributions over many internal states**
  - **Reaction rates account for ground states**
  - **Empirical intermolecular potential is used**

Example: Real Gas Effect Model

ELECTRONIC STATE FOR N ATOM

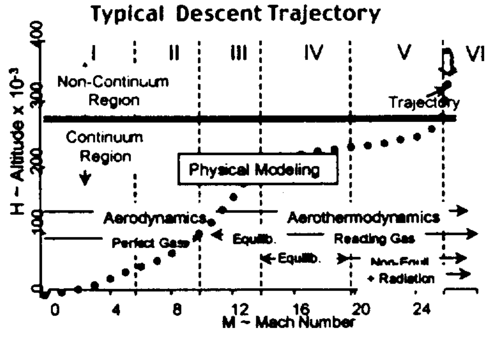


- Proposed Approach
  - Based on more accurate solution of known microscopic equations, develop better macroscopic equations
  - Derive micro eqs and constitutive eqs from Bloltzman eq (inelastic collision)
  - Obtain state-to-state rates and product-state distribution functions

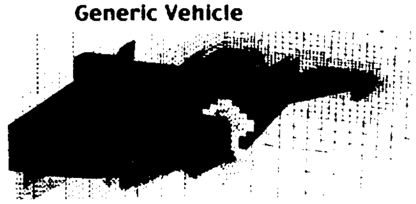
  ⇒Provide macro properties to be used in CFD codes
- Impact
  - The results are more accurate physics-based representation of macroscopic properties (from current curve fitting)
  - Applicable to high-speed planetary reentry

Example of Next Level CFD Problem

Typical Descent Trajectory



Generic Vehicle



- Tough Problems: Physics-Based Scientific Computing + CFD
- Big Impact on Engineering :

  Major contribution in developing a vehicle/system

Examples of Target Problems

- There are many unsolved challenges in developing flow devices
  - Compressor rotational stall
  - Turbopump system in rocket engine
  - Jet noise
  - Maximum lift of high-lift system
  - Rotor-based propulsion system
  - .....

Examples of Target Problems

- There are a wide range of challenging applications in non-aerospace
  Earth simulation
  - Climate prediction
  - Local / regional model

  Biofluid
  - Flow-related problems in human body (e.g. heart, lung, hemodynamics): fundamental understanding / biomedical practics

  Engineering and product development
  - Automobile
  - Naval hydrodynamics
  - Chemical engineering
  - Manufacturing processes
  - ...

Strategy

- Bottom line is "money"
  - Traditional "CFD" research we used to know is probably over.
- Short term
  - Very often, need to satisfy "paying customer"
  - Software for any conventional engineering is most likely available
  - Risk might come from lack of engineering knowledge by CFDers
- Long term
  - Basic research is an investment for the future
  - However, need a "vision" for planning
    Need to identify high payoff areas